150 ptas.



CURSO PRACTICO DEL ORDENADOR PERSONAL, EL MICRO Y EL MINIORDENADOR

Editorial Delta, S.A.

mi computer CURSO PRACTICO
DEL ORDENADOR PERSONAL, EL MICRO Y EL MINIORDENADOR

Publicado por Editorial Delta, S.A., Barcelona, y comercializado en exclusiva por Distribuidora Olimpia, S.A., Barcelona

Volumen I - Fascículo 8

Director: José Mas Godayol
Director editorial: Gerardo Romero
Jefe de redacción: Pablo Parra
Coordinación editorial: Jaime Mardones
Asesor técnico: Roberto Quiroga

Redactores y colaboradores: G. Jefferson, R. Ford, S. Tarditti, A. Cuevas
Para la edición inglesa: R. Pawson (editor), D. Tebbutt (consultant editor), C. Cooper (executive editor), D. Whelan (art editor), Bunch Partworks Ltd. (proyecto y realización)

Realización gráfica: Luis F. Balaguer

Redacción y administración:
Paseo de Gracia, 88, 5.° - Barcelona-8
Tels. (93) 215 10 32 / (93) 215 10 50 - Télex 97848 EDLTE

MI COMPUTER, *Curso práctico del ordenador personal, el micro y el miniordenador,* se publica en forma de 96 fascículos de aparición semanal, encuadernables en ocho volúmenes. Cada fascículo consta de 20 páginas interiores y sus correspondientes cubiertas. Con el fascículo que completa cada uno de los volúmenes, se ponen a la venta las tapas para su encuadernación.
El editor se reserva el derecho de modificar el precio de venta del fascículo en el transcurso de la obra, si las circunstancias del mercado así lo exigieran.

ISBN: 84-85822-83-8 (fascículo) 84-85822-84-6 (tomo 1)
84-85822-82-X (obra completa)
Depósito Legal: B. 52-84

Fotocomposición: Tecfa, S.A., Pedro IV, 160, Barcelona-5
Impresión: Cayfosa, Santa Perpètua de Mogoda (Barcelona) 078403
Impreso en España - Printed in Spain - Marzo 1984

Editorial Delta, S.A., garantiza la publicación de todos los fascículos que componen esta obra.

Distribuye para España: Marco Ibérica, Distribución de Ediciones, S.A., Carretera de Irún, km 13,350. Variante de Fuencarral, Madrid-34.
Distribuye para Argentina: Viscontea Distribuidora, S.C.A., La Rioja 1134/56, Buenos Aires.
Distribuye para Colombia: Distribuidoras Unidas, Ltda., Transversal 93, n.° 52-03, Bogotá D.E.
Distribuye para México: Distribuidora Intermex, S.A., Lucio Blanco, n.° 435, Col. San Juan Tlihuaca, Azcapotzalco, 02400, México D.F.
Distribuye para Venezuela: Distribuidora Continental, S.A., Ferrenquín a Cruz de Candelaria, 178, Caracas, y todas sus sucursales en el interior del país.

Pida a su proveedor habitual que le reserve un ejemplar de MI COMPUTER. Comprando su fascículo todas las semanas y en el mismo quiosco o librería, Vd. conseguirá un servicio más rápido, pues nos permite realizar la distribución a los puntos de venta con la mayor precisión.

Servicio de suscripciones y atrasados (sólo para España)

Las condiciones de suscripción a la obra completa (96 fascículos más las tapas, guardas y transferibles para la confección de los 8 volúmenes) son las siguientes:

a) Un pago único anticipado de 16 690 ptas. o bien 8 pagos trimestrales anticipados y consecutivos de 2 087 ptas. (sin gastos de envío).

b) Los pagos pueden hacerse efectivos mediante ingreso en la cuenta 3371872 de la Caja Postal de Ahorros y remitiendo a continuación el resguardo o su fotocopia a Distribuidora Olimpia (Paseo de Gracia, 88, 5.°, Barcelona-8), o también con talón bancario remitido a la misma dirección.

c) Se realizará un envío cada 12 semanas, compuesto de 12 fascículos y las tapas para encuadernarlos.

Los fascículos atrasados pueden adquirirse en el quiosco o librería habitual. También pueden recibirse por correo, con incremento del coste de envío, haciendo llegar su importe a Distribuidora Olimpia, en la forma establecida en el apartado b).

Para cualquier aclaración, telefonear al (93) 215 75 21.

No se efectúan envíos contra reembolso.



"La voz de su amo"

Los ordenadores ya ocupan un lugar propio en el campo de la música profesional. También se están incorporando pequeños sintetizadores de música en muchos ordenadores personales

Los ordenadores son a la vez divertidos y serios. Y no sirven tan sólo para procesar información o para jugar. También se pueden utilizar como instrumentos musicales para, por decirlo de alguna manera, divertirse con seriedad. El proceso de hacer música artificialmente se denomina síntesis musical.

Los ordenadores también se pueden utilizar para hacer más amena la enseñanza de la música y su concurso resulta bastante más económico que contratar a un profesor particular. En un próximo artículo le explicaremos cómo realizar ese tipo de música con un ordenador personal, pero de momento nos concentraremos en cómo lo hacen los profesionales del medio.

Los instrumentos musicales automáticos siempre han gozado de popularidad y tienen mucho en común con los ordenadores. La pianola, una especie de piano automático que durante el siglo pasado solía adornar las salas de estar de las familias acomodadas, funcionaba mediante un rollo de papel perforado, y las cajas de música llevaban un disco o un tambor metálico con "dientes" que ejecutaban una melodía sobre un peine de metal.

En cierto sentido, hasta los organillos callejeros que funcionaban a manivela eran programables, ya que las melodías que ejecutaban se podían sustituir por otras. No obstante, estos organillos no eran del agrado de Charles Babbage, uno de los padres fundadores de la informática, quien era partidario de que a los organilleros se les prohibiera hacer música en las calles. Éstos, como respuesta, acudían a tocarle melodías bajo su propia ventana.

En la actualidad, en Gran Bretaña, es el Sindicato de Músicos el que está tratando de impedir el uso de dispositivos musicales programables; en mayo de 1982, la sección sindical central de Londres decidió, mediante votación, prohibir su utilización en las sesiones de grabación y en las actuaciones en vivo. Como es evidente, esta preocupación de los sindicatos de músicos nace del hecho de que, dado que estos dispositivos pueden imitar el sonido de muchos instrumentos diferentes y de forma simultánea, con el tiempo los músicos no serían necesarios.

Los sintetizadores electrónicos salieron al mercado hace ya muchos años, pero la introducción de técnicas digitales les ha abierto un campo totalmente nuevo. En vez de tener que manipular clavijas y pulsar botones para producir cada uno de los sonidos, ahora se puede grabar cualquier sonido, analizarlo mediante ordenador en las partes que lo constituyen y reproducirlo en cualquier tono.

El sonido digitalizado es algo así como una fotografía publicada en un periódico; si mira la página desde muy cerca, verá que la imagen se compone de muchos puntos pequeños y separados unos de otros, mientras que la fotografía original (analógica) tiene tonalidades difuminadas de sombras que se mezclan continuamente entre sí. Del mismo modo, el sonido analógico normal se puede descomponer en una secuencia de dígitos. Esta técnica se denomina muestreo.

Los sistemas de este tipo son caros (probablemente los modelos más conocidos y menos sofisticados sean el Fairlight y el Synclavier), pero, puesto que pueden reproducir los sonidos de varios instrumentos musicales, resultan más baratos que contratar a todos los músicos que serían necesarios para este fin.

Con el descenso de los precios de los ordenadores y la progresiva disminución del costo de la memoria, está aumentando la popularidad de las máquinas digitales, aunque falta aún mucho tiempo para que los sintetizadores analógicos desaparezcan por completo. Estos últimos utilizan una técnica denominada "síntesis de sustracción" que, de alguna forma, puede compararse con la manera en que un escultor moldea su estatua en un bloque de mármol. Se comienza con un sonido básico creado electrónicamente y luego se lo hace pasar a través de una serie de procesos electrónicos. Cada proceso modifica o descompone el sonido


Cortesía de EMI

Un hombre, úna orquesta
Los músicos sintetizadores como Klaus Schultz, a quien vemos en la fotografía, están utilizando cada vez más el enorme potencial de sus instrumentos basados en microprocesadores para producir, en vivo, la misma variedad de sonidos para la que hace veinte años se hubiera necesitado toda una orquesta



Ian McKinnell

Efectos especiales
Los sintetizadores de música controlados por ordenadores personales se están haciendo cada vez más populares. Los efectos musicales, que hace diez años sólo existían para los equipos profesionales más costosos, ahora tienen un precio muy asequible. El que vemos en la fotografía puede leer en su memoria música que previamente ha sido codificada en papel en forma de código de barras. Luego el ejecutante la puede reproducir o modificar. Muchos de estos sintetizadores personales se pueden acoplar directamente a un ordenador personal, para aprovechar la pantalla o la memoria adicional. Sin embargo, cada vez son más numerosos los ordenadores personales que llevan incorporada alguna forma de síntesis musical

en sus mínimos componentes, para darle la forma deseada. La síntesis de sustracción estimula a experimentar con diferentes combinaciones de procesos y su técnica resulta sencilla hasta para un principiante.

Por el contrario, todo cuanto se cree con un sintetizador digital ha de planificarse tan cuidadosamente como la construcción de un gran edificio. Ello se debe a que el dispositivo se vale de la *síntesis de adición*: el sonido final se produce mediante la adición de componentes, uno sobre el otro. Uno ha de estar próximo al final del proceso para poder ser capaz siquiera de reconocer mínimamente el sonido. Sin embargo, se puede tomar un sonido convencional, descomponerlo en sus componentes básicos, almacenar éstos en la memoria del ordenador, ya sea en una RAM interna o en un disco o una cinta, y utilizarlos como "ladrillos" para construir el sonido deseado.

Cajas de música
Actualmente existen muchos paquetes para ordenadores personales que permiten aprovechar al máximo la capacidad que éstos poseen para realizar música. En estos paquetes, la visualización en pantalla se puede utilizar para hacer una intepretación visual de la música que se está ejecutando, o para ayudar al músico recién iniciado, que puede pulsar el teclado QWERTY como si fuera el de un piano

Ian McKinnell

Casi tan importante como su aptitud para crear una variedad tan amplia de sonidos individuales es la capacidad que poseen los ordenadores para almacenar secuencias y composiciones musicales. La mayoría de los sintetizadores más vendidos poseen un secuenciador (dispositivo que almacena y recupera las secuencias de sonidos) opcional, aunque es probable que ya lo lleven incorporado. El Fairlight, por ejemplo, puede almacenar en su memoria de disco 30 minutos de sonido de hasta ocho "voces", que se pueden considerar como instrumentos musicales. La capacidad del Synclavier corresponde al doble.

De todo lo anterior se deduce que en los sintetizadores más caros los papeles tradicionalmente asignados al compositor, los músicos y el director se funden en el de una sola persona, que, en gran medida, es, además, un programador.

Si alguna vez ha intentado reproducir a una velocidad mayor una cinta grabada con su voz (o si con ese fin ha reproducido a 45 r.p.m. un disco de 33 r.p.m.), se habrá dado cuenta de que el tono se eleva considerablemente. Una de las características más sorprendentes de los sintetizadores controlados por ordenador es su capacidad para salvar este inconveniente y reproducir una pieza musical a mayor o menor velocidad que la original sin alterar su tono, o, por el contrario, trasponer la melodía a una clave distinta a la misma velocidad.

Incluso es posible tomar, por ejemplo, la parte de una trompeta, duplicarla y, simultáneamente, modificar el sonido para que adopte el de una trompa francesa. Es factible, entonces, tocar ambos instrumentos armónicamente, sea al unísono o de forma alternada Esto se conoce con el nombre de *track bouncing*.

A muchas de estas máquinas se les proporcionan las instrucciones por medio de un lenguaje de composición (el del Synclavier se denomina SCRIPT) que no difiere demasiado del BASIC (también posee números de líneas), aunque tal vez sea un poco más difícil de utilizar. La configuración más relevante es la denominada *compilación invertida*: se ejecuta una pieza musical en el teclado y el ordenador produce un listado en SCRIPT para su composición. Esto equivaldría, si es que uno se pudiera imaginar algo así, a que se pudiera jugar a un juego nuevo creado por uno mismo en la pantalla del ordenador y luego, apretando un botón, ¡se obtuviera un listado completo del programa para este juego!

Si considera que su sentido del ritmo no ha sido todo lo perfecto que deseaba, existe la posibilidad de editar y volver a reproducir el listado en SCRIPT utilizando un teclado convencional (QWERTY) y una pantalla, igual que en BASIC. Uno de los sintetizadores Yamaha tiene un sistema menos flexible pero más atrayente: la melodía que usted haya ejecutado la imprime utilizando el procedimiento musical convencional, es decir, por medio de notas en el pentagrama.

Los sintetizadores también se están introduciendo en el cine. La película *TRON*, producida por Walt

Disney, ha recibido entusiastas críticas, que elogian sus sorprendentes gráficos realizados por ordenador. Mucho menos conocido resulta el hecho de que se emplearon microordenadores para la música y los efectos sonoros. Además de la parte de música "verdadera" ejecutada por la London Philharmonic Orchestra, se grabó y se modificó una gran variedad de sonidos usando un sintetizador Fairlight, incluyendo los ruidos producidos por la aeronave Goodyear y por una nevera.

Fue tal la cantidad de sonidos que se emplearon, que se hubo de mantener actualizado un catálogo completo mediante el paquete de base de datos File Manager 800+, procesado con un ordenador personal Atari 800. Algunas de las voces artificiales de la película también se realizaron mediante un ordenador personal y un dispositivo para síntesis de voz que permitía mezclar voces y música. Atari también les facilitó un paquete que jamás anteriormente se había utilizado fuera de dicha empresa. Consistía en una clase de programa para síntesis de adición que Atari había venido empleando para crear sofisticados efectos de sonido en sus propios programas para juegos y para sus máquinas tragaperras. La impresión general es que la importancia de este "sistema electrónico ensamblador de sonidos" respecto a la creación de sonidos es equiparable a la que tuvo el tratamiento de textos en relación a la creación de textos.

En realidad existe un número bastante elevado de accesorios con los cuales se puede convertir un ordenador personal en un sofisticado sintetizador de música. El Apple es un modelo especialmente popular para este tipo de aplicación, debido a las ranuras para conectar dispositivos adicionales que posee el ordenador en su parte posterior. La finalidad de estos dispositivos es hacer que el ordenador realice todo el trabajo creativo y proporcionar un sintetizador analógico de gran calidad como salida. Otros incluyen un teclado exactamente igual al de un piano.

En el extremo más doméstico del mercado de sintetizadores informatizados, el Casio CT7000 incluye lo que se conoce como *secuenciador polifónico*, el cual, mediante una unidad de cassette y una gran cantidad de memoria RAM, puede funcionar como un sistema de grabación profesional de pistas múltiples. Una grabadora de cassette corriente sólo posee una pista por cada lado, mientras que una estereofónica posee dos. Pero una unidad profesional puede llegar a tener más de 24, de modo que la interpretación de cada instrumento se puede grabar por separado, para luego mezclar todos los sonidos para obtener la banda sonora definitiva. De esta manera, con el CT7000 uno puede crear sus propias sinfonías.

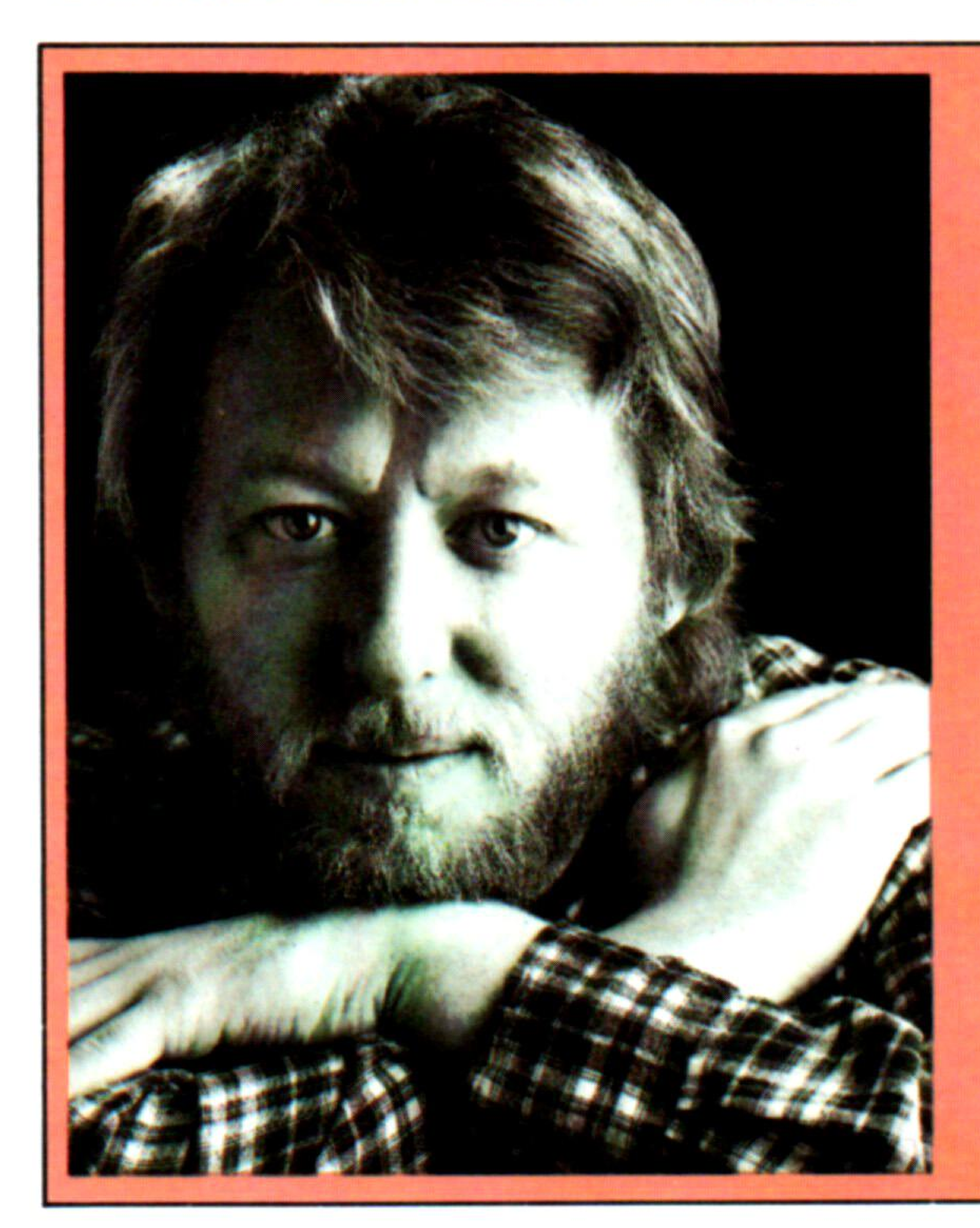
Su antecesor, el CT701, utilizaba un lector de código de barras (véase p. 40) para leer música a partir de los códigos de barras impresos en la memoria de la máquina. Lamentablemente, parecería que no existen medios viables para que el usuario pueda crear códigos de barras impresos de sus propias creaciones, por lo cual este procedimiento sólo se puede emplear con la música preimpresa del sintetizador Casio.

Para triunfar en el mundo de la música pop, ni siquiera se necesita contar con un equipo de este nivel, como lo ha demostrado el grupo alemán Trio, que con su tema *Da Da Da*, para cuya grabación utilizaron un Casiotone VL1 que cuesta 30 libras (unas 6 700 pesetas), obtuvieron más éxito que el que Peter Gabriel consiguiera con un Fairlight de miles de libras.

A pesar de que el VL1 es un dispositivo monofónico (en el cual sólo se puede ejecutar una nota a la vez), es capaz de emular diversos instrumentos musicales y de almacenar una secuencia de notas. También ofrece la posibilidad de cambiar determinadas cualidades de las notas para que el usuario tenga la posibilidad de crear su propio sonido. Esto se conoce como alterar la envoltura ADSR (Attack, Decay, Sustain, Release: atacar, decaer, sostener, soltar). Ahora bien, lo más interesante es que ordenadores personales como el Commodore 64 y el Oric-1 poseen exactamente las mismas características. Y, por supuesto, el solo hecho de que el aparato sea programable en BASIC le da la facultad de contar con un secuenciador incorporado.

Existen en el mercado paquetes de programas para composición musical para muchos de estos ordenadores personales, incluso para los que, comparativamente, poseen configuraciones musicales más sencillas. Algunos de ellos visualizan en pantalla un pentagrama, y la música se compone seleccionando los diferentes tipos de notas musicales con la ayuda de una palanca de mando o de un lápiz óptico, y colocándolas en el pentagrama. Al pulsar el botón de "disparo" de la palanca de mando, o dar alguna orden igualmente sencilla, la composición musical empieza a ser ejecutada. Otra alternativa consiste en que la visualización represente el teclado de un piano; en este caso las notas también se seleccionan con el lápiz óptico o con la palanca de mando, o bien con el teclado del ordenador.

Los efectos musicales pueden programarse en BASIC sin que sea necesario utilizar un paquete de este tipo. Al igual que las configuraciones para gráficos, el procedimiento exacto varía considerablemente de una máquina a otra, así como la sofisticación de las órdenes en BASIC creadas con este fin. El Dragon, por ejemplo, posee una sola voz, pero dispone de una orden PLAY que inicia una secuencia de notas, que se digitan como la antigua notación de la escala musical: de la A a la G (de *la* a *sol*, según la notación actual). El Commodore 64, por el contrario, tiene incorporadas en su hardware muchas y sofisticadas configuraciones musicales, pero no posee órdenes en BASIC listas para utilizar, como el Dragon. Próximamente le enseñaremos cómo realizar música con un ordenador.



Martin Rushent

Productor de grupos como Altered Images y Dexy's Midnight Runners, Martin Rushent es uno de los vanguardistas de los sintetizadores musicales controlados por ordenador. En su estudio del distrito rural de Berkshire hay no menos de nueve sistemas diferentes, que representan la situación actual de este arte. Cada uno de estos sistemas utiliza un procedimiento de programación diferente y distintos códigos para las notas de una escala musical. Puesto que no ha sido factible conectarlos a todos entre sí para formar un único sistema gigante, Rushent, para trasponer las piezas musicales de uno a otro sistema, utiliza un programa que escribió él mismo en un pequeño ordenador personal.

Len Lewis

Bien direccionada

La CPU ha de localizar las instrucciones y los datos almacenados en miles de bytes de memoria del ordenador. Le revelamos lo que sucede en la CPU al ejecutar las instrucciones de un programa

La CPU recibe sus instrucciones y sus datos desde las direcciones situadas en la memoria del ordenador, colocando las patillas de dirección en el código binario requerido por la localización de memoria y leyendo luego el contenido de la localización en la CPU a través del bus de datos. Expuesto de esta manera, el proceso no parece ofrecer mayores dificultades; sin embargo, nada más lejos de la realidad, por cuanto en la práctica esta operación es bastante más complicada.

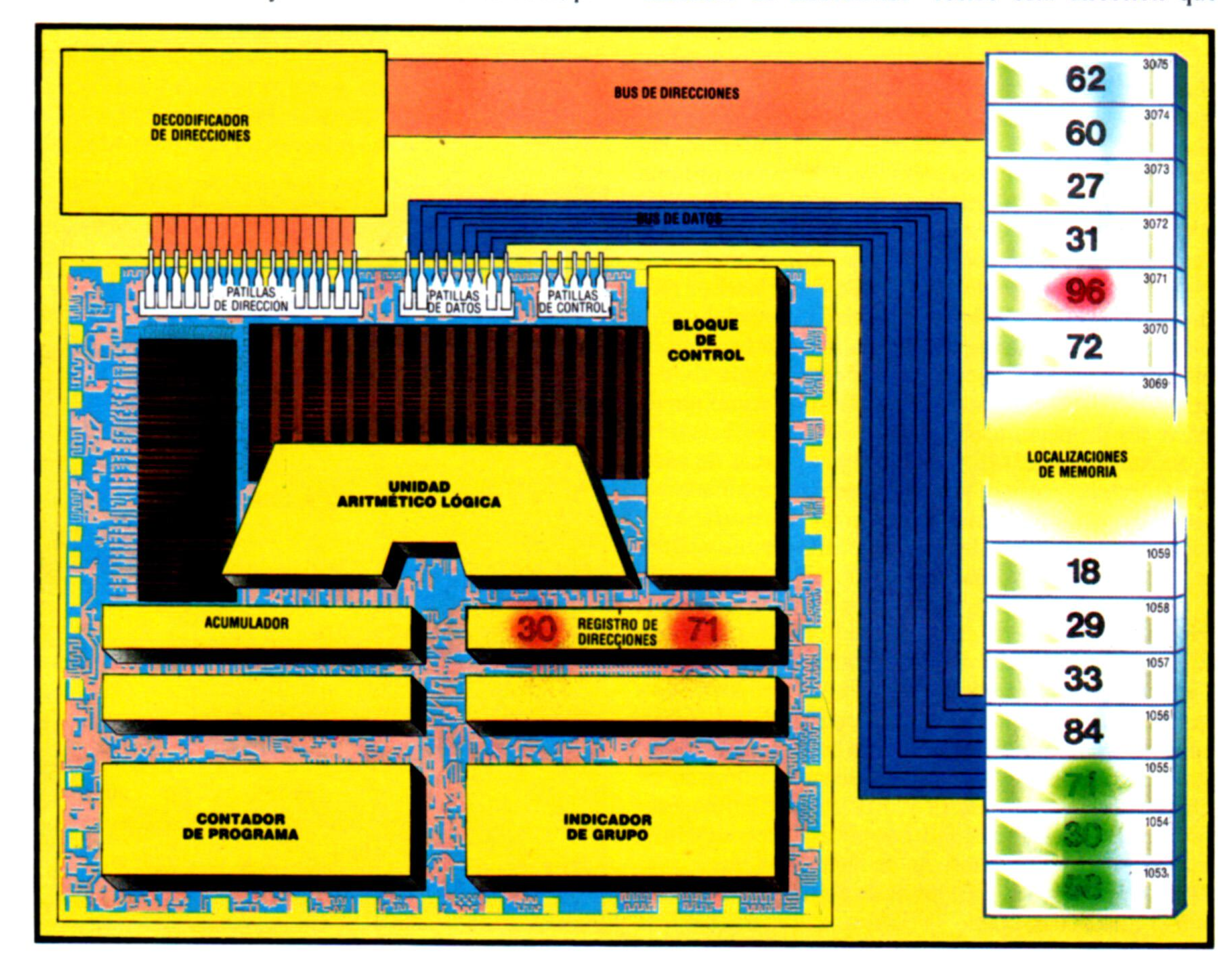
El problema reside en que los bytes (códigos binarios de ocho bits) de cualquiera de las miles de celdas de memoria del ordenador pueden ser instrucciones que le digan a la CPU que realice algo, o bien información que la CPU ha de manipular de alguna forma. ¿Cómo sabe la CPU cuáles bytes son instrucciones y cuáles son datos?

Reconociendo los códigos

En primer lugar, veamos qué es una "instrucción". Una instrucción es un código, en binario, que hace que en el interior de la CPU se lleve a cabo una secuencia específica de operaciones. De modo que el código 00111010, en caso de que la CPU lo reconozca como una instrucción y no tan sólo como un dato, podría hacer que la CPU direccionara los dos próximos bytes de la memoria, leyera los datos que contienen, colocara esos datos en un "registro de direcciones" especial, situara las patillas de dirección en el mismo número, fuera hasta la localización de memoria recientemente direccionada, a continuación colocara el contenido de dicha localización en el bus de datos y, para finalizar, cargara aquellos contenidos en el acumulador de la CPU.

Todo esto, expresado en palabras, puede resultar confuso; pero lo que acabamos de describir es uno de los procedimientos de direccionamiento de memoria que utiliza la CPU del popular Z80. El gráfico ilustra el proceso completo de tomar un byte de información de la memoria y colocarlo en la CPU. Supongamos que la CPU ya sabe que el próximo byte de la memoria al cual accederá será una instrucción (y no información) y que este byte está en la localización de memoria 1053. (Todos los números utilizados en esta ilustración están en notación decimal.) Esta dirección, la 1053, se colocará en el bus de direcciones. En binario es 0000010000011101. Las 16 patillas de dirección están en posición "encendido" o "apagado" de modo tal que correspondan a este número. Cuando el "decodificador de direcciones" recibe esta dirección que

Cadenas de acontecimientos
Aun la operación más sencilla de la CPU implica muchas etapas. Las instrucciones, también llamadas "códigos de operación", se le leen a la CPU desde la memoria. Estas instrucciones son decodificadas por el bloque de control y hacen que se lleven a cabo operaciones específicas. En este ejemplo, se lee la instrucción 58 de la localización de memoria 1053. Esta instrucción determinada hace que se produzca la siguiente cadena de acontecimientos: el byte de la siguiente localización de memoria (1054) será leído y almacenado en una mitad del registro de direcciones de 16 bits de la CPU. El byte de la localización siguiente (1055) será leído y almacenado en la otra mitad. Ahora estos dos bytes representan la dirección (en algún otro lugar de la memoria) donde se han almacenado unos datos. El contenido del registro de direcciones se coloca en el bus de direcciones, de modo que la próxima localización de memoria a la que se acceda será la dirección 3071. El contenido de esta dirección se pone en el bus de datos y se le lee a la CPU. Este byte (el 96 en nuestro ejemplo) es colocado entonces en el acumulador de la CPU, donde permanecerá hasta que una instrucción posterior lo requiera. El bus de direcciones regresará entonces a su anterior dirección +1, de manera que ahora se dirigirá a la localización 1056. La CPU sabe que, sea lo que fuere lo que contenga esa localización, ha de ser una instrucción, y que se repetirá una secuencia de operaciones similares. En este ejemplo la siguiente instrucción es la 84, que es interpretada por el bloque de control para "complementar" o invertir los bits del acumulador. Dado que 84 es una instrucción "de un byte", la CPU sabe que el byte de la próxima localización de memoria, la 1057, también será una instrucción



viene en el bus de direcciones, la "decodifica" y enciende una, y sólo una, de sus líneas de salida. Ésta es la línea que se encarga de seleccionar la localización de memoria 1053.

La próxima etapa consiste en colocar el contenido de esta dirección, que es 58, o 00111010 en binario, en el bus de datos y "cargarlo" en la CPU. Aquí, como la CPU está esperando una instrucción, el byte es interpretado por el bloque de control y hace que se realice una secuencia muy precisa de operaciones. Esta instrucción en particular especifica que los dos próximos bytes de memoria contendrán 16 bits para ser utilizados como localización de memoria, y que el contenido de esta localización se ha de cargar en el acumulador de la CPU. No bien la CPU reconoce esta instrucción, sabe que los dos próximos bytes de memoria especificarán una dirección y que el contenido de esa dirección se habrá de cargar en el acumulador. En consecuencia, sabe que no recibirá otra instrucción desde la memoria hasta después que se hayan efectuado estas operaciones, y que la próxima instrucción estará en la localización 1056.

La instrucción que estamos utilizando en el ejemplo hace que el bus de direcciones se incremente en 1, de manera que la próxima localización de memoria que se direcciona es la 1054. El contenido de esta localización se coloca entonces en el bus de datos y se carga en la CPU. Esta vez, sin embargo, se coloca en la mitad de un registro de dirección. Después de hacerlo, la CPU vuelve a incrementar el bus de direcciones de modo que ahora direccione la ubicación 1055. El contenido de esta localización se coloca en el bus de datos e, igualmente, se carga en la CPU, si bien en esta ocasión pasa a almacenarse en la otra mitad del registro de direcciones.

Transfiriendo números

La próxima etapa (recuerde que todas estas acciones se producen automáticamente a consecuencia de la instrucción original) consiste en que los números del registro de direcciones se transfieren al bus de direcciones. Estos números, como podemos ver, son 3071. Por lo tanto, la localización de memoria que se está direccionando ahora es la 3071. Esta dirección (en binario 0000101111111111) es decodificada por el decodificador de direcciones y selecciona la celda de memoria 3071. El contenido de esta localización, 96 (en binario 01100000), se coloca en el bus de datos y se carga en la CPU. Esta vez, sin embargo, la información se colocará en el acumulador de la CPU. Después de efectuada esta operación, el bus de direcciones se establecerá en 1056 y allí la CPU esperará encontrar otra instrucción.

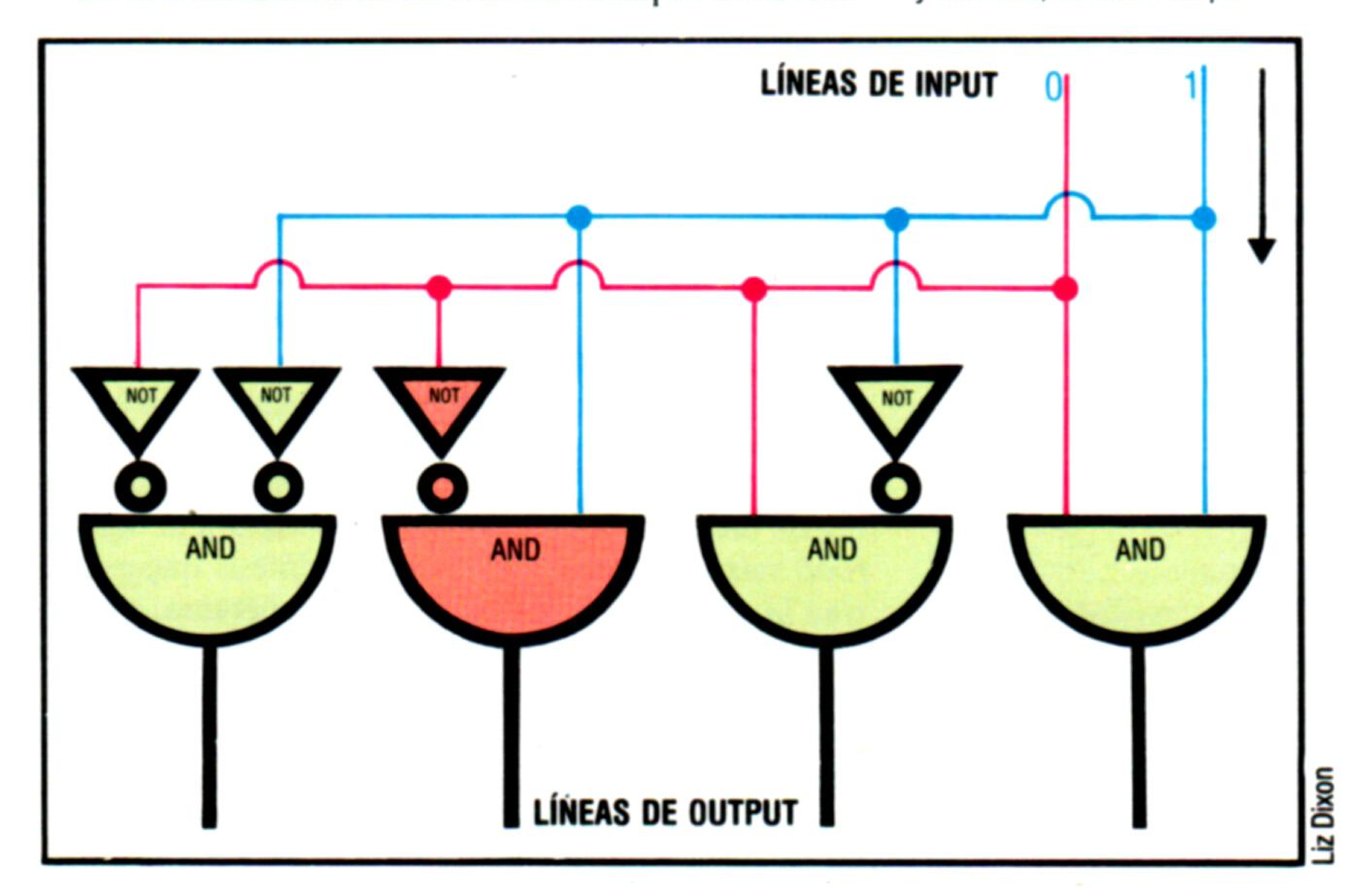
Ahora que la CPU posee en su acumulador una información determinada, ¿qué clase de instrucción se podría esperar a continuación? Podría ser prácticamente cualquiera; las CPU son capaces de reconocer desde docenas hasta centenares de instrucciones, según la CPU. Pero supongamos que lo que deseábamos era invertir los datos del acumulador. Invertir significa cambiar cada uno por un cero y cada cero por un uno. La instrucción para hacerlo estaría localizada en la dirección 1056. En nuestra CPU imaginaria, el código para esta instrucción sería 84. Cuando la CPU recibiera este número, se invertirían los datos del acumulador. El número que estaba en el acumulador era 96 (01!00000 en binario). La instrucción para invertir haría que éste se sustituyera por el número binario 10011111. La instrucción para invertir un número del acumulador es una instrucción "de un byte", de modo, pues, que nuevamente la CPU sabría que el contenido de la siguiente localización de memoria, 1057, será otra vez una instrucción y no un dato.

Este procedimiento de direccionar una localización de memoria para recuperar un dato sólo es uno de los diversos métodos de que dispone el programador. Los bytes de instrucción específicos que hemos utilizado en el ejemplo (58 para cargar el acumulador y 84 para invertir los contenidos del acumulador) son instrucciones para nuestra CPU hipotética, pero el mismo principio se aplica para todos los otros chips de microprocesadores. La única diferencia estriba en que se emplean códigos distintos para las diversas instrucciones y que cada versión de CPU posee su propio "juego de instrucciones", que difiere ligeramente del de las demás.

Las localizaciones de I/O (Input/Output) también deben poseer direcciones exclusivas, pero los principios en virtud de los cuales la CPU las direcciona son los mismos. Por lo general, en los microprocesadores de ocho bits sólo ocho de las líneas de dirección están disponibles para el direccionamiento de I/O, de manera que el número máximo de direcciones de I/O es 256. No obstante, esto es más que suficiente para la mayoría de las aplicaciones de los ordenadores pequeños.

La decodificación de direcciones siempre es necesaria, de manera que el dispositivo seleccionado por la CPU (ya sea una localización de memoria o una localización de I/O) está especialmente activo cuando todas las demás localizaciones de memoria o de I/O están inactivas. Este proceso se denomina *activación*. Cuando hay que decodificar una pequeña cantidad de líneas de dirección, se pueden utilizar chips de puertas lógicas simples para que realicen la decodificación. El gráfico ilustra el principio del decodificador de línea de dos a cuatro. Este tipo de decodificación sencilla se emplea con frecuencia para seleccionar dispositivos de I/O. Sin embargo, a medida que va aumentando el número de líneas de dirección, la complejidad del circuito de decodificación se incrementa de manera considerable. Cuando es preciso proceder a seleccionar individual y exclusivamente 65 536 localizaciones de memoria separadas, lo normal es que la mayor parte de la decodificación de direcciones se lleve a cabo en el interior de los chips de memoria.

Decodificación de direcciones
Las 16 líneas que constituyen el bus de direcciones son capaces de identificar de modo exclusivo cualquiera de las 65 636 localizaciones de memoria separadas. La combinación de unos y ceros del bus de direcciones se decodifica en los decodificadores de direcciones. Parte de la decodificación la efectúan los decodificadores de direcciones, compuestos por puertas lógicas simples en chips montados en el circuito; gran parte de la decodificación la efectúan circuitos equivalentes situados dentro de los propios chips de memoria. La ilustración muestra cómo se pueden decodificar dos líneas de dirección para seleccionar uno, y sólo uno, de cuatro chips



Totalmente funcional

El BASIC posee funciones incorporadas; ello significa que gran parte de la programación ya ha sido hecha para usted. El saber utilizarlas le permitirá disponer de un mayor potencial informático

Supongamos que en uno de sus programas desea averiguar la raíz cuadrada de un número. La puede calcular de diversas maneras. La forma más elemental y menos satisfactoria sería crear una tabla de valores de raíz cuadrada y utilizarla para obtener el valor deseado para un número determinado. Es probable que haya aprendido en la escuela la forma de confeccionar una tabla de este tipo. Un método alternativo consiste en utilizar la "función" raíz cuadrada, que incorporan la mayoría de las versiones de BASIC. Aquí es este lenguaje el que se encarga de la aritmética de la operación, sin que el programador deba preocuparse por ella. Veamos cómo funciona:

```
10 REM ESTE PROGRAMA BUSCA LA RAIZ
   CUADRADA
20 REM DE UN NUMERO
30 PRINT "DE ENTRADA AL NUMERO CUYA RAIZ
   CUADRADA"
40 PRINT "DESEA HALLAR"
50 INPUT N
60 LET A = SQR(N)
70 PRINT "LA RAIZ CUADRADA DE"; N; "ES"; A
80 END
```

Digite este corto programa y verá que, en efecto, le proporcionará la raíz cuadrada de cualquier número que digite. Estudiemos las reglas relativas a la forma de utilizar esta función de "raíz cuadrada".

Una "función" en BASIC es generalmente una palabra de orden (en este caso SQR, abreviatura de *square root*: raíz cuadrada) seguida de paréntesis que encierran la expresión a operar. En este programa, N es la entrada del número desde el teclado. Es el número cuya raíz cuadrada deseamos hallar. La línea 60 dice "asignemos la raíz cuadrada de N a la variable A". La línea 70 imprime el valor de A.

La expresión entre paréntesis se denomina *argumento* de la función y no siempre ha de ser una variable: es igualmente posible utilizar números reales. Digite lo siguiente y vea qué sucede cuando lo ejecuta:

```
10 PRINT SQR(25)
20 END
```

Comprobará que esto funciona exactamente igual. De la misma manera, podemos incluir entre los paréntesis argumentos más complicados. Pruebe con este ejemplo:

```
10 LET A = 10
20 LET B = 90
30 LET C = SQR(A + B)
40 PRINT C
50 END
```

Este pequeño programa se puede acortar combinando las líneas 30 y 40 de la siguiente manera:

```
10 LET A = 10
20 LET B = 90
30 PRINT SQR(A + B)
40 END
```

Las funciones se deben considerar como programas cortos incorporados al BASIC, disponibles para que el programador los emplee en cualquier momento. La mayoría de las versiones de BASIC ofrecen una cantidad bastante importante de funciones, así como la posibilidad de que el programador defina algunas nuevas para utilizarlas dentro de un programa. Más adelante veremos la forma de hacerlo. Ahora estudiaremos algunas de las otras funciones disponibles comúnmente. Éstas presentan dos variedades: las funciones numéricas, en las cuales el argumento (la parte encerrada entre paréntesis) es un número, una variable numérica o una expresión numérica, y las funciones en serie, en las cuales el argumento es una serie de caracteres o una expresión alfanumérica. Veamos primero algunas de las funciones numéricas.

Con anterioridad, en la página 77, ejecutamos un programa que calculaba el número de azulejos que se necesitaban para revestir las paredes de una habitación. Un pequeño *bug* de este programa consistía en que la respuesta no podía incluir fracciones decimales de un azulejo. La ejecución de este programa podría dar 988,24 como posible resultado. En ocasiones como ésta deseamos disponer de algún procedimiento para redondear la respuesta en el número completo más próximo. En matemáticas, los números completos se denominan *enteros* y una de las funciones del BASIC es "devolver" la parte entera de cualquier número. Funciona así:

```
10 PRINT "DE ENTRADA A UN NUMERO QUE
   CONTENGA FRACCION DECIMAL"
20 INPUT N
30 PRINT "LA PARTE ENTERA DEL NUMERO ES";
40 PRINT INT(N)
50 END
```

Si ejecuta este programa y da entrada a 3,14, el programa imprimirá en pantalla:

```
LA PARTE ENTERA DEL NUMERO ES 3
```

Por supuesto, si se trata de azulejos necesitaríamos luego sumarle 1 a la respuesta, para asegurarnos de comprar más cantidad de la requerida y no menos.

En otra ocasión podríamos desear averiguar el "signo" de un número para ver si es negativo, cero o positivo. Para ello, la mayoría de las versiones de BASIC incorporan una función SGN. Ensayemos con el siguiente ejemplo:

```
10 PRINT "DE ENTRADA A UN NUMERO"
20 INPUT N
```

```
30 LET S = SGN(N)
40 IF S = -1 THEN GOTO 100
50 IF S = 0 THEN GOTO 120
60 IF S = 1 THEN GOTO 140
100 PRINT "EL NUMERO ERA NEGATIVO"
110 GOTO 999
120 PRINT "EL NUMERO ERA CERO"
130 GOTO 999
140 PRINT "EL NUMERO ERA POSITIVO"
150 GOTO 999
999 END
```

Si observa los valores "devueltos" a S en la línea 30 por la función SGN (éstos se comparan en las líneas 40, 50 y 60), verá que los valores son tres. Se devuelve −1 si el argumento entre paréntesis era un número negativo, 0 si el argumento era cero y 1 si el argumento era un número positivo. La utilización de la función SGN en la línea 30 ahorra varias líneas de programación. Podríamos haber escrito:

```
IF N < 0 THEN LET S = -1
IF N = 0 THEN LET S = 0
IF N > 0 THEN LET S = 1
```

La acción que se consigue realizar mediante el empleo de una función de BASIC siempre se puede obtener mediante la programación normal, pero el empleo de una función supone un ahorro de tiempo, de espacio y de esfuerzo de programación.

A continuación reseñamos unas pocas funciones numéricas más. ABS devuelve el valor "absoluto" de un número. El valor absoluto de un número es el de su valor real pero sin su signo. Por lo tanto, el valor absoluto de −6 es 6. Probemos:

```
10 LET X = -9
20 LET Y = ABS(X)
30 PRINT Y
40 END
```

MAX busca el valor máximo de dos números. Por ejemplo:

```
10 LET X = 9
20 LET Y = 7
30 LET Z = X MAX Y
40 PRINT Z
50 END
```

MIN es similar a MAX, pero averigua el valor más pequeño de dos números. Veamos el proceso:

```
10 PRINT "DE ENTRADA A UN NUMERO"
20 INPUT X
30 PRINT "DE ENTRADA A OTRO NUMERO"
40 INPUT Y
50 LET Z = X MIN Y
60 PRINT Z
70 END
```

Observe que estas dos últimas funciones poseen dos argumentos en lugar de uno, y que no es necesario encerrarlos entre paréntesis. La mayoría de las versiones de BASIC poseen también algunas otras funciones numéricas, incluyendo LOG para encontrar el logaritmo de un número, TAN para descubrir la tangente, COS para encontrar el coseno y SIN para hallar el seno. Más adelante veremos algunas de las maneras en que se pueden utilizar estas funciones "trigonométricas".

El BASIC posee, asimismo, varias funciones incorporadas que operan con variables alfanuméricas. En nuestro programa para seleccionar un nombre (p. 135) empleamos algunas de ellas, pero en aquella ocasión no estudiamos con detención la forma en que funcionaban. Ahora veremos con mayor detalle aquellas funciones en serie y algunas más.

Una de las funciones en serie más útiles es LEN. Ésta cuenta el número de caracteres de una serie de ellos encerrada entre comillas dobles o el número de caracteres asignados a una variable alfanumérica. Probemos con el siguiente ejemplo:

```
10 LET A$ = "COMPUTER"
20 LET N = LEN(A$)
30 PRINT "EL NUMERO DE CARACTERES DE LA
   VARIABLE ES"; N
40 END
```

¿Por qué necesitaríamos saber, alguna vez, de cuántos caracteres se compone una variable alfanumérica? Para ver por qué, dé entrada y ejecute este breve programa creado para construir un "triángulo de nombre". En primer lugar imprimirá la primera letra de una palabra, luego la primera y la segunda, después la primera, la segunda y la tercera, y así sucesivamente hasta imprimir la palabra completa.

```
5 REM IMPRIME UN 'TRIANGULO DE NOMBRE'
10 LET A$ = "JONES"
20 FOR L = 1 TO 5
30 LET B$ = LEFT$(A$,L)
40 PRINT B$
50 NEXT L
60 END
```

Ahora ejecute este programa. ¿Se imagina qué es lo que imprimirá? Se leerá algo similar a esto:

```
J
JO
JON
JONE
JONES
```

Este corto programa utiliza la función LEFT$ para extraer los caracteres de una variable. LEFT$ lleva dos argumentos. El primero especifica la variable y el segundo (que viene precedido por una coma) especifica el número de caracteres a extraer de la variable, empezando desde el lado izquierdo de ésta. A A$ se le había asignado la serie "JONES", de modo que LEFT$(A$,1) "devolvería" la letra J. LEFT$(A$,2) devolvería las letras JO. El breve programa anterior emplea un índice, L, que abarca desde 1 hasta 5, de manera que el segundo argumento de la función LEFT$ asciende desde 1 hasta 5 cada vez que se efectúa el bucle. Sabemos exactamente cuántos caracteres había en la palabra que deseábamos imprimir (JONES), por lo cual fue sencillo decidir que 5 sería el límite máximo del bucle FOR-NEXT. Pero ¿qué haríamos si no supiéramos de antemano cuántos caracteres habrá de tener el bucle?

Aquí entra en juego la función LEN. Esta función toma como argumento una variable (entre comillas dobles) o una variable alfanumérica. A continuación le proporcionamos algunos ejemplos para que vea cómo funciona LEN:

```
10 REM PROGRAMA PARA COMPROBAR LA
   FUNCION 'LEN'
20 PRINT LEN ("COMPUTER")
30 END
```

Al ser ejecutado, este programa imprimirá 8. Ha contado el número de caracteres que componen la palabra COMPUTER y ha devuelto este valor. Hagamos lo mismo pero de un modo ligeramente diferente:

```
10 REM BUSCANDO LA LONGITUD DE UNA VARIABLE
20 LET A$ = "MI COMPUTER"
30 LET L = LEN(A$)
40 PRINT L
50 END
```

Este programa, al ser ejecutado, habría de imprimir en pantalla 11. En esta variable hay 11 caracteres y no 10. Recuerde que, en lo que atañe al ordenador, el espacio que separa una palabra de otra es también un carácter. Ahora apliquemos la función LEN en una modificación de nuestro programa anterior para imprimir un "nombre triangular":

```
10 REM ESTE PROGRAMA IMPRIME UN 'TRIANGULO
   DE NOMBRE'
20 PRINT "DIGITE UN NOMBRE"
30 INPUT A$
40 LET N = LEN(A$)
50 FOR L = 1 TO N
60 LET B$ = LEFT$(A$,L)
70 PRINT B$
80 NEXT L
90 END
```

Cada vez que se ejecute este bucle, el valor de L se incrementará desde 1 hasta N (que es la longitud del nombre en la variable). Si ahora se da entrada al apellido SIMPSON, la línea 40 equivaldrá a LET N = LEN ("SIMPSON"), de manera que N se establecerá en 7. La primera vez que se efectúe el bucle, la línea 50 establecerá L en 1 y la línea 60 equivaldrá a LET B$ = LEFT$ ("SIMPSON", 1), de modo que a B$ se le asignará un carácter de la variable, empezando por la izquierda. Este carácter es S.

La segunda vez que se efectúe el bucle, L se establecerá en 2, de modo que la línea 60 equivaldrá a LET B$ = LEFT$ ("SIMPSON", 2). Ésta tomará los dos primeros caracteres de la variable y los asignará a la variable alfanumérica B$. Por lo tanto, B$ contendrá SI.

La función LEN descubrió que los caracteres de la variable SIMPSON eran 7 y le asignó este valor a la variable N, de manera que la última vez que se efectúe el bucle a B$ se le asignarán los 7 caracteres de la variable y se imprimirá la variable completa.

LEFT$ posee una función compañera, RIGHT$, que toma los caracteres de la variable alfanumérica exactamente de la misma forma pero comenzando por la derecha.

Por último, estudiaremos otra función alfanumérica, que también empleábamos en nuestro programa de clasificación de nombres. Se trata de INSTR y se utiliza para encontrar la localización de la primera aparición de una variable específica (denominada *subvariable*) dentro de una variable. En el programa de clasificación de nombres se utilizaba INSTR para localizar la posición del espacio entre el nombre de pila y el apellido. Funciona de la siguiente manera:

```
10 LET A$ = "GUARDABOSQUE"
20 LET P = INSTR(A$, "BOSQUE")
30 PRINT P
40 END
```

Antes de dar entrada a este programa y de ejecutarlo, intente predecir el valor que se imprimirá para P. Recuerde que INSTR localiza la posición inicial de la primera vez que se produce la subvariable dentro de la variable. Si la variable es GUARDABOSQUE, la posición inicial de la subvariable BOSQUE será 7, pues la B de BOSQUE es la séptima letra de GUARDABOSQUE. Algunas versiones de BASIC carecen de la función INSTR, pero poseen una función similar denominada INDEX. Ésta es la forma de utilizar INSTR (o INDEX) para localizar un espacio dentro de una variable:

```
10 REM BUSCANDO LA POSICION DE UN ESPACIO EN
   UNA VARIABLE
20 LET A$ = "ORDENADOR PERSONAL"
30 LET P = INSTR(A$, " ")
40 PRINT P
50 END
```

Observe que el segundo argumento de la función INSTR (línea 30) es " ". Las comillas encierran un espacio: el carácter que se ha de buscar. El programa imprimirá 10 como el valor de P, dado que el espacio se halla en el décimo lugar de la variable. Intente deducir qué se imprimiría si se modificara la línea 30 del siguiente modo:

```
LET P = INSTR(A$, "C")
```

Por último, una función muy práctica que se utiliza con la sentencia PRINT. Veamos lo que sucede al ejecutar este programa:

```
10 PRINT "ESTA LINEA NO ESTA SANGRADA"
20 PRINT TAB(5); "ESTA LINEA ESTA SANGRADA"
30 END
```

¿Puede ver lo que sucedió? La segunda línea se empezó a imprimir cinco espacios a la derecha del margen izquierdo. TAB es análoga al tabulador de una máquina de escribir. Aquí le ofrecemos otro breve programa que emplea la función TAB:

```
10 REM UTILIZANDO LA FUNCION TAB
20 PRINT "INTRODUZCA EL VALOR DE TAB"
30 INPUT T
40 LET W$ = "TABULACION"
50 PRINT TAB(T); W$
60 END
```

Ahora puede volver al programa de clasificación de nombres, en la página 136, y comprobar cómo se emplean en él algunas de estas funciones.

Ejercicios

■ **Bucles 1** ¿Qué se imprimirá al ejecutar este programa?

```
10 LET A = 500
20 FOR L = 1 TO 50
30 LET A = A −1
40 NEXT L
50 PRINT "EL VALOR DE A ES"; A
```

Complementos al BASIC



En el Spectrum reemplazar por el número de TAB.



El Spectrum no dispone de ninguna de estas órdenes, pero usted puede crear sus propias versiones de las mismas mediante la sentencia DEF FN; de modo que agregue a su programa las siguientes líneas:



```
9900 DEF FN L$(X$,N) = X$(TO N)
9910 DEF FN R$(X$,N) =
   = X$(LEN(X$) - N + 1 TO)
9920 DEF FN M$(X$,P,N) = X$(P TO
   P + N - 1)
```

Ahora



FN L$(X$,N) sustituye a LEFT$(X$,N)
FN M$(X$,P,N) sustituye a MID$(X$,P,N)
FN R$(X$,N) sustituye a RIGHT$(X$,N)



No disponen de esta función el Spectrum, el VIC-20, el Commodore 64 y el Oric-1, pero es posible sustituirla. Supongamos que la línea original fuera:

```
20 LET P = INSTR(A$,"BOSQUE")
```

Reemplácela por

```
20 LET X$ = A$: LET Z$ = "BOSQUE"
   GOSUB 9930: P = U
```

y agregue estas líneas:

```
9929 STOP
9930 LET U = 0: LET X = LEN(X$):
   LET V = LEN(Z$)
9940 FOR W = 1 TO L - V + 1: IF
   MID$(X$,W,V) = Z$ THEN LET U = W
9950 IF U <> 0 THEN LET W =
   = L - V + 1
9960 NEXT W: RETURN
```

En el Spectrum, sustituya MID$(X$,W,V), en la línea 9940, por FN M$(X$,W,V) y consulte el "Complementos al BASIC" anterior

■ **Bucles 2** ¿Qué se verá en pantalla si se ejecuta este programa?

```
10 REM
20 REM ESTE ES UN BUCLE DE TIEMPOS
30 REM COMPROBAR CUANTO DURA
40 REM
50 PRINT "START"
60 FOR X = 1 TO 5000
70 NEXT X
80 PRINT "STOP"
90 END
```

■ **Bucles 3** ¿Qué resultado se imprimirá si ejecuta este programa y, al requerírsele, digita el número 60?

```
10 PRINT "PIENSE UN NUMERO Y DIGITELO"
20 INPUT N
30 LET A = 100
40 FOR L = 1 TO N
50 LET A = A + 1
60 NEXT L
70 PRINT "AHORA EL VALOR DE A ES"; A
80 END
```

■ **Bucles 4** ¿Qué sucedería al ejecutar este programa?

```
10 PRINT "ME GUSTA EL BASIC"
20 GOTO 10
30 END
```

■ **Bucles 5** ¿Qué se vería en pantalla si se ejecutara este programa?

```
10 FOR Q = 1 TO 15
20 PRINT "ESTOY ALGO TONTO"
30 NEXT Q
40 END
```

■ **Read-Data 1** ¿Qué resultado se imprimiría?

```
10 READ X
20 READ Y
30 READ Z
40 PRINT "COMPARAMOS LA SENTENCIA 'READ'"
50 DATA 50,100,20
60 PRINT X + Y + Z
```

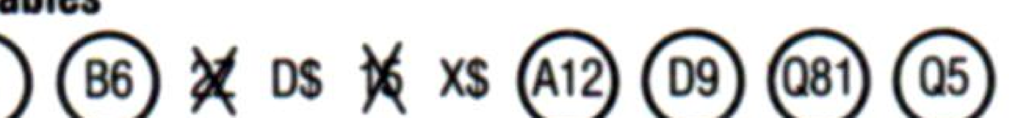
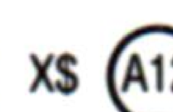
■ **Read-Data 2** ¿Qué se imprimiría en pantalla si se ejecutara este programa?

```
100 FOR L = 1 TO 10
110 READ X
120 PRINT "X = "; X
130 NEXT L
140 DATA 1,2,3,5,7,11,13,17,19,23
```

Las respuestas, en el próximo capítulo.

Respuestas a los "Ejercicios" de las páginas 136 y 137

Variables

(A) (B6) ~~2X~~ D$ ~~1X~~ X$ (A12) (D9) (Q81) (Q5) ~~6X~~ H$

Aritmética 1

```
10 LET B = 6
20 PRINT B
```

Aritmética 2

```
10 LET A = 5
20 LET B = 7
30 LET C = 9
40 LET D = A + B + C
50 PRINT D
```

Aritmética 3

17

Aritmética 4

25
25

Comparaciones 1

5

Comparaciones 2

601 (los enteros se suponen)

Comparaciones 3

10000

Print 1

```
PRINT "EL VALOR DE T ES";T
```

Print 2

```
640 PRINT "LO SIENTO, PERO SU PUNTUACION DE";S; "ES
      DEMASIADO BAJA"
```

Print 3

Se trataba de un error deliberado. El punto y coma al final de la línea producirá un error de sintaxis cuando se ejecute. El programa debe decir:

```
200 LET A$ = "¿MI COMPUTER?"
210 LET B$ = "¿LE GUSTA?"
220 PRINT B$;A$
```

Y entonces el resultado sería:
¿LE GUSTA MI COMPUTER?

Entrada 1

6

Entrada 2

POR FAVOR DIGITE SU NOMBRE
HOLA (SU NOMBRE) SOY SU ORDENADOR

Tenga en cuenta que las respuestas a las "variables" serán diferentes en algunas máquinas que no admiten más de un carácter alfabético (es decir, que no admiten sufijo numérico)

Jupiter Ace

El único ordenador personal de precio económico cuyo lenguaje de programación estándar es Forth en lugar de Basic; todo un desafío para programadores ambiciosos

El Jupiter Ace es una máquina para entusiastas de la informática y es uno de los pocos ordenadores cuyo lenguaje estándar no es el BASIC. Sus diseñadores habían formado parte del equipo del ZX Spectrum y decidieron producir un ordenador que a ellos mismos les agradara usar.

El lenguaje incorporado es el FORTH y ésta es la característica distintiva de esta máquina. Pero su hardware es lo suficientemente barato como para que cualquier persona interesada en aprender FORTH se pueda comprar el Ace en lugar de perfeccionar el ordenador que ya posea. El manual que viene con el ordenador constituye una excelente iniciación al lenguaje FORTH.

La línea del Ace se asemeja mucho a la de los modelos Sinclair. De hecho, su carcasa es del mismo plástico blanco y delgado que poseía el primer ordenador

Introducción al Forth

Versión en BASIC

```
100 REM UN PROGRAMA EN BASIC PARA IMPRIMIR
    'SHAZAM!'
110 FOR X = 1 TO 6
120 PRINT "SHAZAM!"
130 NEXT X
140 END
    RUN
```

Versión en FORTH

```
(UN PROGRAMA EN FORTH PARA IMPRIMIR 'SHAZAM!')
:SHOUT ." SHAZAM! " ; .
:CHORUS 6 0 DO SHOUT LOOP;
 CHORUS
```

Estos dos programas harán exactamente lo mismo, pero la versión en BASIC semeja una receta, mientras el programa en FORTH ¡parece un sistema de escritura para adivinos!

El lenguaje FORTH comienza con un grupo de palabras órdenes (denominadas diccionario), y tiene la particularidad de que se le pueden agregar nuevas palabras. En nuestro programa en FORTH agregamos dos nuevas palabras al diccionario: SHOUT se define como una serie de caracteres a imprimir y CHORUS se define como una mezcla de "primitivas" (palabras predefinidas en el diccionario) y la nueva palabra SHOUT.

El FORTH también posee una memoria (denominada *stack*) y capacidad de procesar los números que ella contenga. El programa en FORTH efectúa las mismas operaciones aritméticas y lógicas que el programa en BASIC, pero no las lleva a cabo a través de expresiones algebraicas, sino manipulando el stack.

El lenguaje FORTH puede llegar a ser exasperante, algo así como un "cubo de Rubik"; no obstante, es innegable que constituye un eficaz método de programación y toda una nueva forma de pensar.

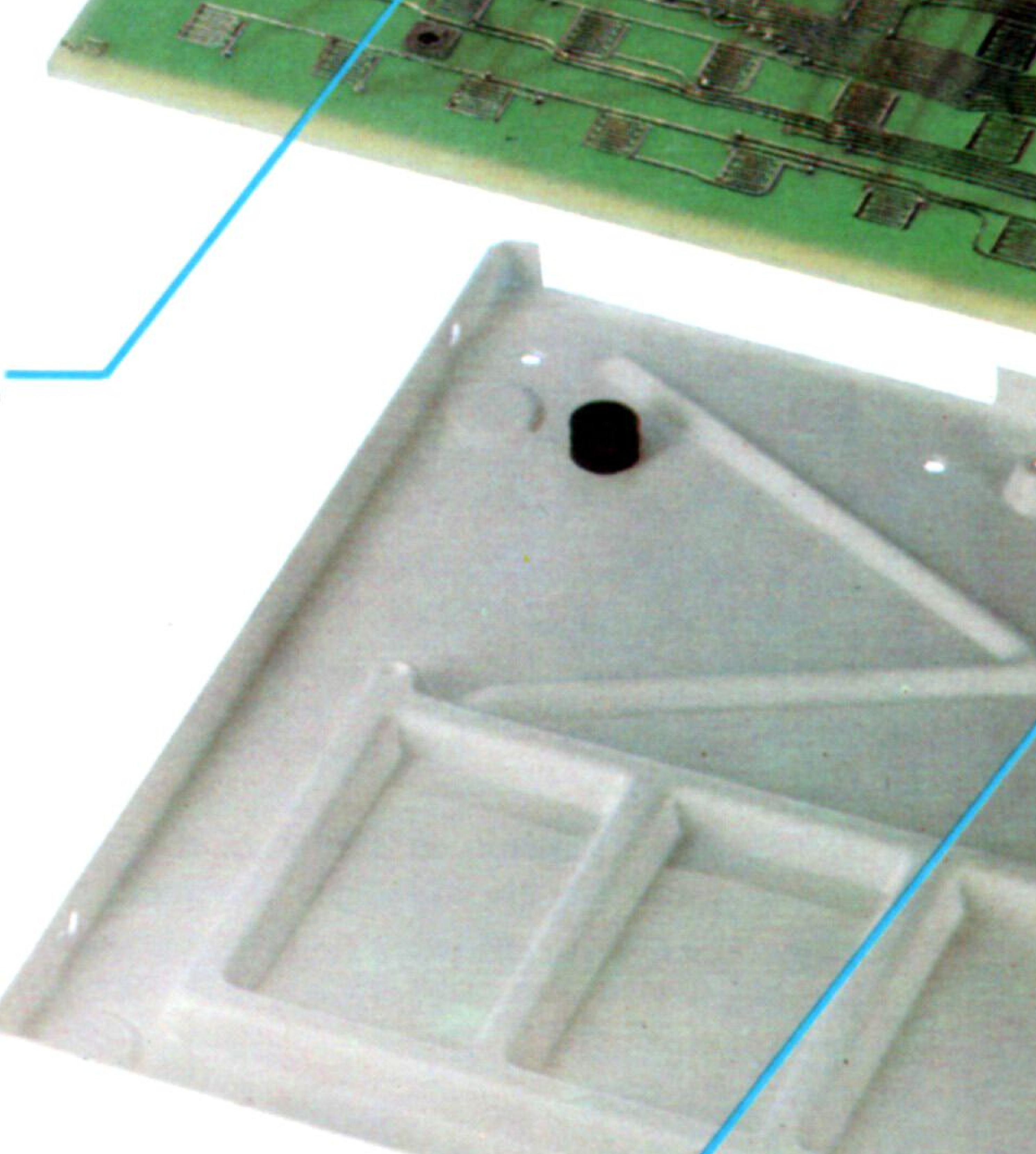
Lo que verdaderamente le confiere al FORTH todo su poder es su capacidad para definir y utilizar nuevas órdenes. En efecto, el usuario puede confeccionar el lenguaje de programación a la medida de la aplicación que está realizando. El FORTH es especialmente adecuado para programar dispositivos automáticos de uso doméstico, ya que el informático puede crear su propio diccionario de órdenes: MOVE (mover), FETCH (alcanzar), FIND (hallar), FOLLOW (seguir) y RETURN (retornar), por ejemplo



Teclado del Jupiter Ace
Las 40 teclas móviles están moldeadas sobre una sola lámina de plástico, prácticamente como en el Sinclair Spectrum. Todas las teclas de la hilera superior poseen tres funciones, a las que se accede mediante las teclas SHIFT y SYMBOL SHIFT. Además, se pueden utilizar siete caracteres para gráficos (más *space*) para construir diagramas y gráficos sencillos

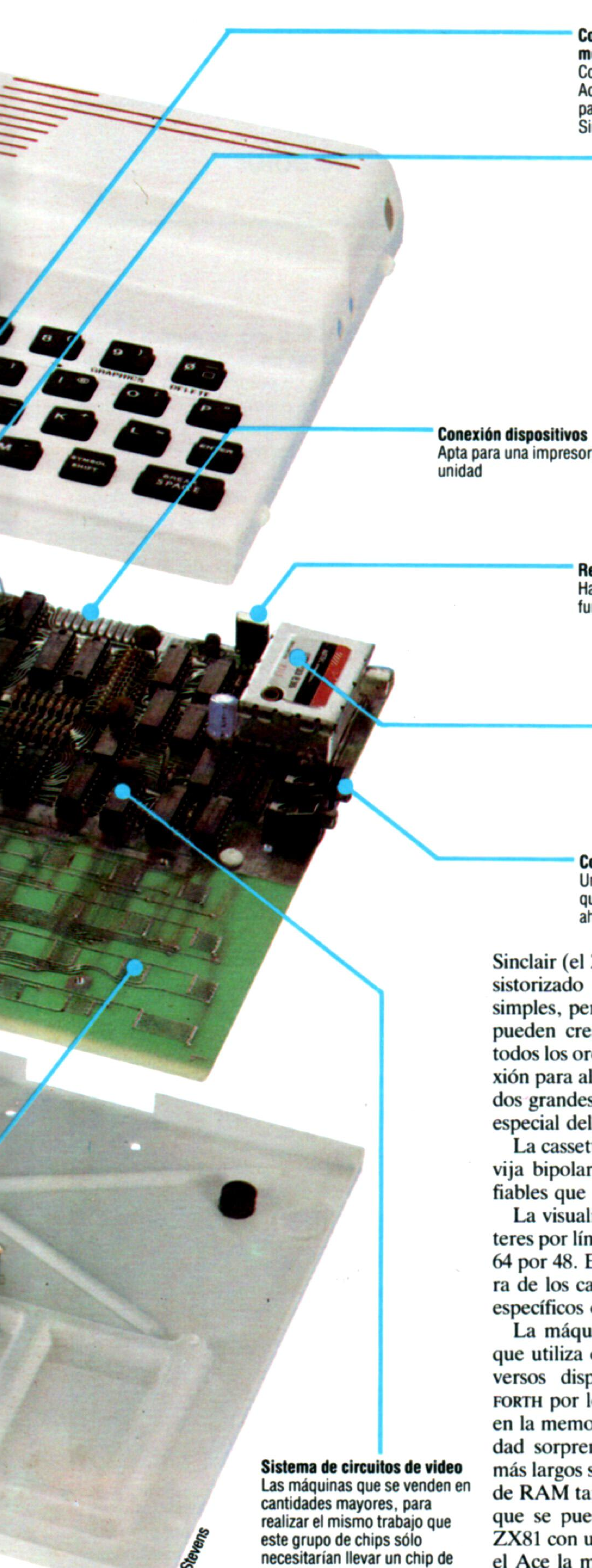
Microprocesador
Lleva un Z80A, lo cual no es sorprendente, ya que los diseñadores de la máquina también fueron responsables del diseño del Sinclair Spectrum





Altavoz
Es un dispositivo piezoeléctrico transistorizado (bastante parecido a los zumbadores de los relojes digitales), que se puede utilizar para generar sonidos simples

Rellenos del teclado
Cuando se pulsa una tecla, el material plástico especial, que es conductor de la electricidad, hace contacto entre dos juegos de pistas metálicas situadas sobre el tablero impreso del circuito



Conexión para ampliación de memoria
Con un adaptador adecuado, el Ace también puede utilizar los paquetes de ampliación para el Sinclair ZX81

Lenguaje FORTH
Para retener el lenguaje se han empleado dos unidades EPROM de 4 Kbytes. El pedido de unidades ROM ha de efectuarse por millares de unidades, de modo que para volúmenes de fabricación más pequeños se utilizan unidades EPROM. La parte superior de cada EPROM posee una cubierta protectora; de lo contrario, cualquier luz ultravioleta borraría su contenido

Conexión dispositivos
Apta para una impresora u otra unidad

Reloj
Hace que el microprocesador funcione a 1 MHz

Modulador de RF
Proporciona una salida en blanco y negro sólo para televisor

Conexión cassette
Uno de los sistemas más fiables que se han desarrollado hasta ahora

Sistema de circuitos de video
Las máquinas que se venden en cantidades mayores, para realizar el mismo trabajo que este grupo de chips sólo necesitarían llevar un chip de diseño especial

Sinclair (el ZX80). Posee un pequeño zumbador transistorizado que puede producir una gama de tonos simples, pero con un mayor conocimiento del Ace se pueden crear sonidos más complejos. Al igual que todos los ordenadores Sinclair, el Ace posee una conexión para alimentación eléctrica, otra para televisión y dos grandes conexiones para cualquier otro elemento especial del equipo.

La cassette cuenta con una conexión sencilla de clavija bipolar, una de las interfaces para cassette más fiables que puede poseer un ordenador.

La visualización es en blanco y negro con 32 caracteres por línea, si bien su resolución para gráficos es de 64 por 48. El usuario puede volver a diseñar cualquiera de los caracteres para crear símbolos matemáticos específicos o formas determinadas para juegos.

La máquina goza ya de gran aceptación debido a que utiliza el lenguaje FORTH, y existen a la venta diversos dispositivos accesorios. Los programas en FORTH por lo general son muy cortos, de manera que en la memoria estándar, de 3 Kbytes, cabe una cantidad sorprendente de ellos. Para escribir programas más largos se necesita más memoria. Existen paquetes de RAM tanto de 16 Kbytes como de 48 Kbytes, aunque se pueden emplear los paquetes RAM para el ZX81 con un adaptador. Con éste se pueden usar con el Ace la mayoría de los accesorios Sinclair, siempre que se disponga de diversos tipos de entrada y salida, como puertas en paralelo y conexiones seriales.

JUPITER ACE

DIMENSIONES
215 × 190 × 30 mm

PESO
246 g

CPU
Z80A

VELOCIDAD DEL RELOJ
1 MHz

MEMORIA
RAM de 3 Kbytes ampliable exteriormente a 51 Kbytes; ROM de 8 Kbytes

VISUALIZACION EN VIDEO
Blanco y negro, líneas de texto de 32 × 22, gráficos de 64 × 48

INTERFACES
Conexiones televisión cassette y alimentación eléctrica (9 v); dos grandes conexiones: la primera tiene líneas completas de dirección y datos del procesador; la segunda posee líneas de datos y algunas de selección

LENGUAJE SUMINISTRADO
FORTH

OTROS LENGUAJES DISPONIBLES
Ninguno

VIENE CON
Unidad de alimentación eléctrica en el enchufe principal, cables para cassette y televisión

TECLADO
Relleno plástico similar al del Sinclair Spectrum pero más blando y menos preciso. Las teclas se han de pulsar exactamente por el centro. Todas las teclas disponen de repetición automática y dos teclas de cambio permiten producir todos los códigos de ASCII

DOCUMENTACION
Se trata, con toda seguridad, del mejor manual para ordenadores pequeños que se haya realizado, y haría enrojecer de vergüenza a la gran mayoría de los proveedores de ordenadores mayores. Su autor es el mismo que escribió los manuales del ZX81 y del Spectrum. En 180 páginas ofrece una introducción al FORTH y una descripción completa del Ace, y siempre con el apoyo de numerosos ejemplos. Dispone de una lista de contenidos, de cuatro apéndices que proporcionan rápidas referencias de todas las facilidades disponibles, y de un índice

Gráficos con vida

Las memorias de gran capacidad permiten producir con ordenadores personales imágenes plenas de color y movimiento

Una de las características más sorprendentes de los ordenadores personales es su capacidad para producir gráficos y visualizaciones en movimiento, comúnmente denominadas imágenes animadas. En la mayoría de los microordenadores, el usuario puede trazar puntos individuales, dibujar líneas y círculos y cambiar los colores del fondo y de los primeros planos.

Para los juegos de simulacro y de acción rápida, debemos ser capaces de simular el movimiento. La manera más sencilla de conseguirlo consiste en producir una serie de imágenes fijas, una después de la otra. Esto se debe realizar con la rapidez suficiente como para crear la ilusión de movimiento. Las imágenes de televisión se producen mediante un procedimiento similar al descrito.

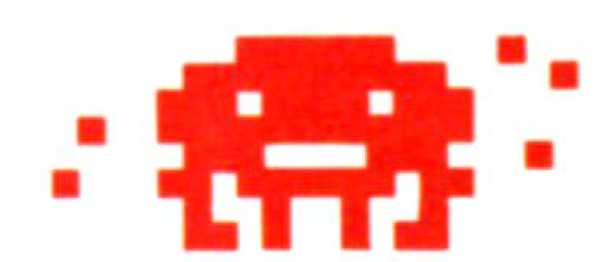
Rapidez de acción

Otra forma de crear la ilusión de movimiento consiste en imprimir un carácter, borrarlo y volverlo a imprimir en una posición ligeramente alejada de la original. Para que el movimiento fluya uniformemente es imprescindible que la distancia entre un paso y otro sea mínima. Asimismo, el tiempo invertido en producir la forma y en borrarla debe ser lo más corto posible.

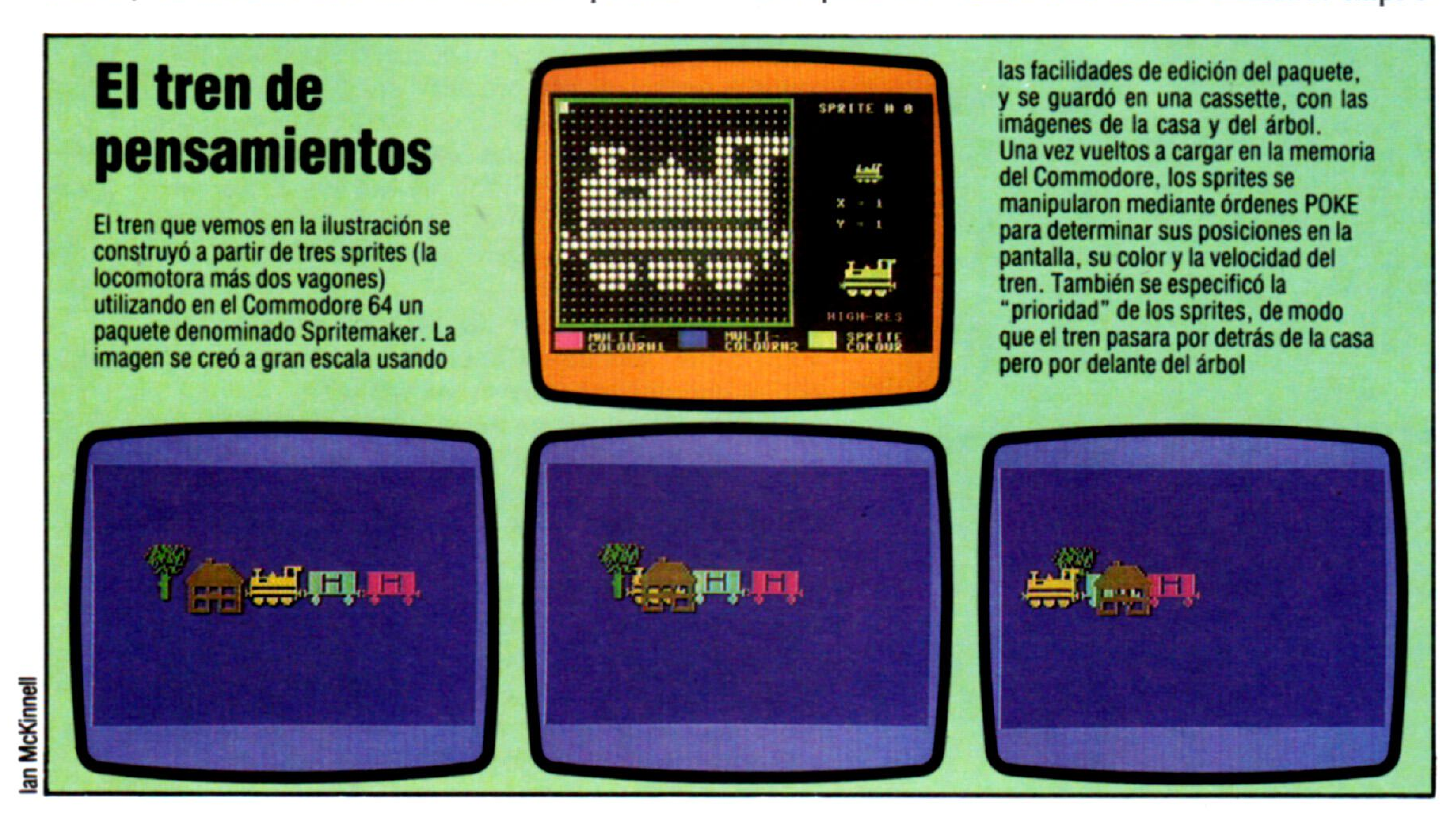
La utilización del BASIC para producir animación da como resultado un movimiento lento. Una forma de superar este inconveniente es recurrir al lenguaje ensamblador, método que requiere mucha práctica, atención y cuidado si desea obtener visualizaciones libres de parpadeos. Sumado a ello, se plantea el problema adicional de controlar los caracteres individuales a medida que aumenta su número en la pantalla.

Diversos ordenadores, en particular el Commodore 64, el Sord M5, el Texas Instruments TI99/4A y la gama Atari, superan este inconveniente y proporcionan una animación que utiliza las mismas técnicas que emplean las máquinas de juegos recreativos accionados con monedas. Esta técnica se conoce como *gráficos sprite*.

Los sprites son "objetos" o formas que pueden ser desplazados, independientemente unos de otros, a través de la pantalla. Esto se logra, sencillamente, cambiando el contenido de un par de direcciones de memoria, que especifican las coordenadas X e Y (las posiciones izquierda-derecha y arriba-abajo). Típicamente, X puede abarcar entre 0 y 255, e Y entre 0 y 191. Algunos sistemas permiten incluso especificar la velocidad y la dirección del movimiento de cada sprite, y el ordenador hace todo el resto.



Los sprites normalmente se realizan utilizando chips o un sistema de circuitos de hardware especiales dentro del ordenador. Es posible comprar software para otros ordenadores con el fin de obtener resultados similares, pero por lo general esta solución resulta poco satisfactoria.



El tren de pensamientos

El tren que vemos en la ilustración se construyó a partir de tres sprites (la locomotora más dos vagones) utilizando en el Commodore 64 un paquete denominado Spritemaker. La imagen se creó a gran escala usando las facilidades de edición del paquete, y se guardó en una cassette, con las imágenes de la casa y del árbol. Una vez vueltos a cargar en la memoria del Commodore, los sprites se manipularon mediante órdenes POKE para determinar sus posiciones en la pantalla, su color y la velocidad del tren. También se especificó la "prioridad" de los sprites, de modo que el tren pasara por detrás de la casa pero por delante del árbol

Ian McKinnell

PLANO 0

PLANO 1

PLANO 2

PLANO 3

PLANO 4

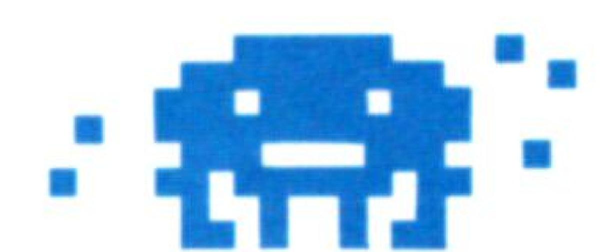
Mark Watkinson

Los sprites se pueden mover a distintas velocidades, ya que cada uno de ellos está situado en lo que se conoce como un *plano de sprite*. Una escena de gráficos en pantalla se compone, por tanto, de una serie de planos apilados uno detrás del otro, si bien a simple vista uno los percibe en la pantalla como si se tratara de un solo plano.

El efecto tridimensional se obtiene haciendo pasar los sprites por delante o por detrás de los demás. Los sprites se numeran de 0 hasta el máximo número disponible que, por ejemplo, en el Sord M5 es 32. Cuando dos sprites se superponen se visualiza el que posee el número más bajo. De manera que si los sprites se ordenan cuidadosamente, se puede obtener con suma facilidad un efecto tridimensional: por ejemplo, si se hace pasar un tren por delante de un árbol, éste se oscurecerá a medida que el tren vaya avanzando. Por otra parte, el mismo tren se puede neutralizar parcialmente si se lo hace pasar por detrás de una casa que posea un número más bajo.

Utilizando la gama de colores que ofrezca el ordenador, cada sprite se puede colorear de forma individual. A veces se puede ampliar o reducir un sprite si se cambia el contenido de una dirección de memoria.

Con todo, es en los programas de juegos donde los sprites demuestran toda su valía; la *detección de colisiones* es una configuración particularmente útil. Cuando se superponen dos o más sprites (por ejemplo, cuando su misil entra en contacto con la nave espacial enemiga), el sistema puede ser programado para que salte hasta otra parte del programa con el fin de crear los gráficos para una explosión.



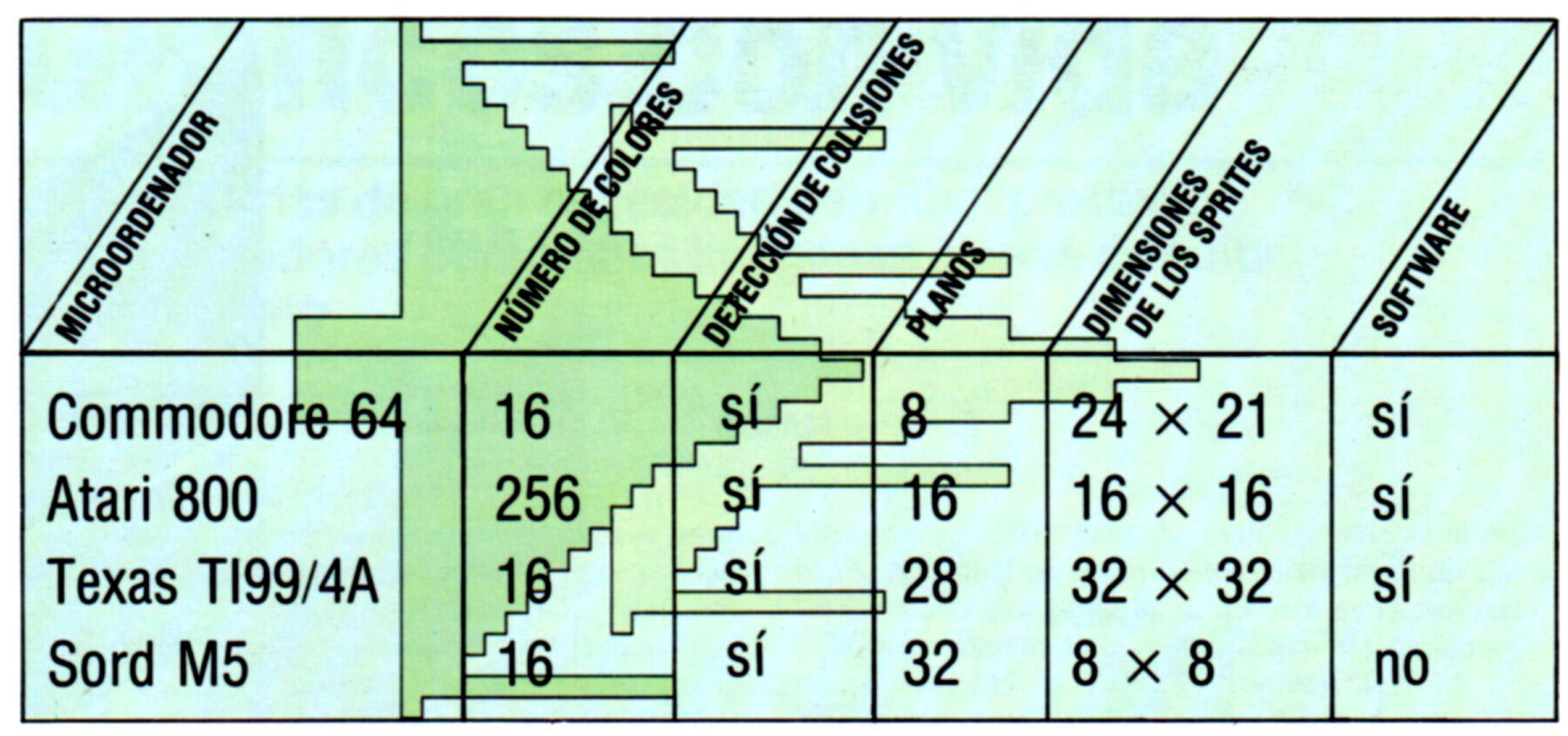
Antes de utilizar los sprites es preciso crearlos, y el modo de hacerlo se asemeja mucho a la forma en que se diseña un nuevo carácter. Las letras del alfabeto, los números y los símbolos especiales para gráficos están almacenados dentro del ordenador, en un chip que se conoce como *generador de caracteres*.

Los caracteres, como ya hemos mencionado en otro apartado de esta obra, por lo general se construyen en una matriz de puntos (o *pixels*) de ocho por ocho. Las dimensiones máximas de los sprites varían de una máquina a otra, pero generalmente su ancho y su alto corresponden a los de varios caracteres. En el Commodore 64, por ejemplo, la dimensión máxima es 24 pixels de ancho por 21 de alto.

La mejor forma de realizar un sprite consiste en tra-

En diferentes planos
Utilizando los gráficos sprite, se puede crear una compleja imagen al situar los elementos componentes en distintos planos, uno detrás de otro. La inmensa ventaja que ofrece este sistema es que los objetos de los diferentes planos se pueden mover de forma absolutamente independiente respecto a los demás, aunque en la pantalla se perciban como si se tratara de un solo plano. El programador les asigna a los planos un orden de prioridad, de modo que cuando en la pantalla se superponen dos sprites, el que posee la prioridad más alta es el que se ve por delante del otro, obteniéndose así un efecto tridimensional
La mayoría de los ordenadores que disponen de gráficos sprite también pueden disponer que el programa se interrumpa cada vez que entran en contacto dos sprites, para crear la consiguiente explosión o aumentar el puntaje del jugador

Dimensiones del sprite
Sólo cuatro de los microordenadores personales más populares disponen de gráficos sprite: el Commodore 64, el Sord M5, el TI99/4A y la gama Atari.
En el caso del Atari, los 16 colores que normalmente se ofrecen pasan a ser 256, ya que cada color posee 16 grados de "luminosidad".
Cuando se utilizan para juegos, a menudo es interesante saber cuándo dos sprites se tocan (por ejemplo, al chocar dos naves espaciales), y por ello se proporciona una configuración para "detección de colisiones".
El efecto tridimensional se crea colocando cada sprite en un plano diferente; lo mejor para un sistema es que disponga de la mayor cantidad de planos posible. Las dimensiones máximas de cada sprite se expresan en pixels. Los sprites se pueden ampliar o reducir, y desplazar por la pantalla.
Para simplificar el proceso de construcción de los sprites existen paquetes especiales de "programas de utilidad"



MICROORDENADOR	NÚMERO DE COLORES	DETECCIÓN DE COLISIONES	PLANOS	DIMENSIONES DE LOS SPRITES	SOFTWARE
Commodore 64	16	sí	8	24 × 21	sí
Atari 800	256	sí	16	16 × 16	sí
Texas TI99/4A	16	sí	28	32 × 32	sí
Sord M5	16	sí	32	8 × 8	no

zar una cuadrícula que represente las dimensiones máximas, y construir la forma deseada rellenando los cuadrados adecuados. Ya se sabe que los ordenadores trabajan utilizando solamente unos y ceros (véase p. 28). Los cuadrados en color se deben representar como unos y los cuadrados en blanco como ceros.



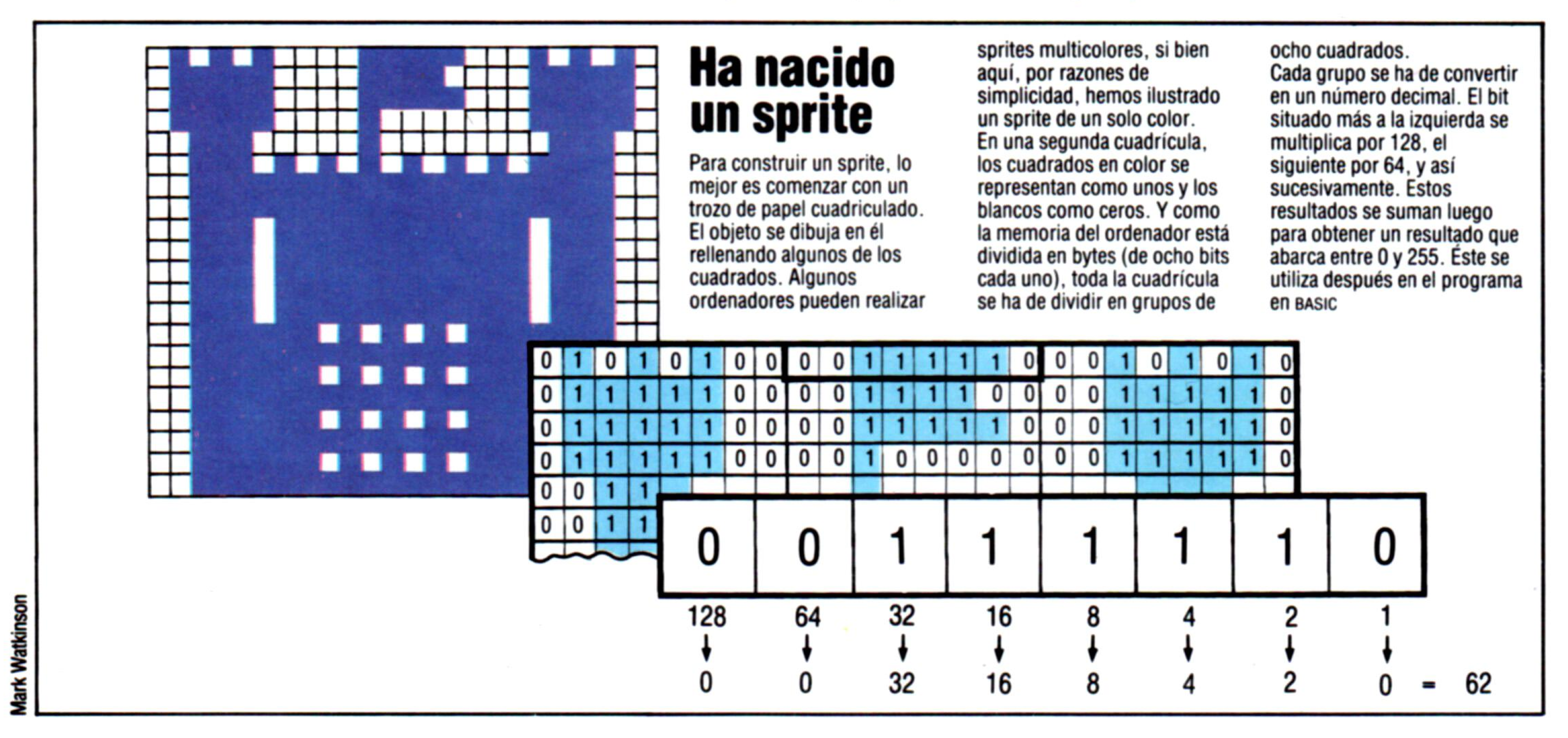
El ordenador maneja la mayor parte de la información en forma de bytes (grupos de ocho bits). En el manual del ordenador se le explicará cómo se debe separar por grupos de ocho la cuadrícula entera de puntos que compone un sprite. Cada byte, a su vez, ha de convertirse en un número decimal, entre 0 y 255 como máximo, antes de que se lo pueda utilizar para un programa en BASIC. Esto se consigue multiplicando el dígito binario (bit) situado más a la izquierda por 128, el siguiente por 64, y así sucesivamente. Luego se suman los resultados.

La serie de números decimales resultante define por completo el diseño del sprite. Estos números se colocan en direcciones de memoria empleando un programa en BASIC; el procedimiento exacto a seguir varía de una máquina a otra. Luego el ordenador requiere que se le proporcionen instrucciones acerca de en qué lugar de la memoria puede hallar las especificaciones para cada uno de los sprites requeridos.

Entonces se puede realizar lo que se desee utilizando órdenes simples para especificar la posición normal de cada sprite en la pantalla, para cambiar su color, agrandarlo o reducirlo y detectar cuándo se superponen dos sprites o más.

Existen paquetes de software, denominados *programas de utilidad*, para la mayoría de las máquinas con sprites. Estos programas logran que el proceso de creación de la imagen resulte menos tedioso. Visualizan la cuadrícula en pantalla a gran escala y permiten dibujar la imagen simplemente moviendo el curso intermitente a través de la cuadrícula. Toda la aritmética se efectúa de forma automática y luego los resultados se colocan en los bytes adecuados. Por último, la cuadrícula desaparece y aparece en pantalla el propio sprite, listo para ser manipulado.

Los sprites han revolucionado los gráficos de los ordenadores personales al proporcionar un método sencillo y eficaz de producir visualizaciones con movimiento rápido y a todo color.



Ha nacido un sprite

Para construir un sprite, lo mejor es comenzar con un trozo de papel cuadriculado. El objeto se dibuja en él rellenando algunos de los cuadrados. Algunos ordenadores pueden realizar sprites multicolores, si bien aquí, por razones de simplicidad, hemos ilustrado un sprite de un solo color.
En una segunda cuadrícula, los cuadrados en color se representan como unos y los blancos como ceros. Y como la memoria del ordenador está dividida en bytes (de ocho bits cada uno), toda la cuadrícula se ha de dividir en grupos de ocho cuadrados.
Cada grupo se ha de convertir en un número decimal. El bit situado más a la izquierda se multiplica por 128, el siguiente por 64, y así sucesivamente. Estos resultados se suman luego para obtener un resultado que abarca entre 0 y 255. Éste se utiliza después en el programa en BASIC

Mark Watkinson



Steve Wozniak



Apple Computer Inc.

De bromista autodidacta de la electrónica a multimillonario: cómo se crearon los populares ordenadores Apple

En Estados Unidos, el gran público conoce más a Steve Wozniak por la puesta en escena de multitudinarios conciertos de rock al aire libre que por ser el hombre que construyó, él solo y en el garaje de su casa, los Apple I y II. Pero en el mundo de la informática se le considera el "genio de la electrónica" que más ha hecho por simplificar y popularizar el microordenador. Las máquinas de Wozniak fueron las primeras en disponer de color, gráficos, teclado y video como configuraciones estándar. Y el Apple II es tan popular que ya se han vendido un millón de unidades.

La meteórica ascensión de Steve Wozniak, desde ingeniero en el garaje de su casa hasta multimillonario en dólares, parece un moderno cuento de hadas. Wozniak nació y creció en el ahora famoso lugar de nacimiento del chip, Silicon Valley (California). Su padre, ingeniero de profesión, le enseñó la ley más elemental de la electrónica, la ley de Ohm, pero a partir de ahí Wozniak aprendió electrónica por sí mismo.

De niño ocupaba su tiempo jugando con componentes electrónicos y aplicando su genio técnico en hacer travesuras en la escuela. Una vez construyó un dispositivo electrónico llamado caja azul, idea que concibió inspirándose en el personaje de una revista que era un proscrito de la tecnología. El dispositivo podía imitar ciertos tonos a través del sistema telefónico. Estos tonos especiales eran la indicación de que la persona que telefoneaba había insertado en la caja de monedas del teléfono la cantidad de dinero necesaria. Esto le permitió a Wozniak hacer llamadas telefónicas gratis a todo el mundo. Habló con amigos de Inglaterra, y por este procedimiento llamó incluso ¡al Papa!

Wozniak jamás estudió la carrera de ingeniería; en la escuela era un alumno excelente en matemáticas y en electrónica, pero abandonó los estudios. Su primer trabajo fue como técnico en la inmensa corporación norteamericana Hewlett Packard, donde diseñó calculadoras. Pero ellos le dijeron que no estaba capacitado para hacer lo que él realmente deseaba: diseñar ordenadores. De modo que empezó a trabajar por su cuenta, especialmente por la noche, y diseñó un microordenador que la Hewlett Packard rechazó. Wozniak no se desalentó; dejó la compañía, y con Steve Jobs, condiscípulo suyo en la escuela (y compañero de travesuras), construyó y vendió 50 unidades de la máquina que había diseñado. Así nació el Apple I. Bautizaron con el nombre de Apple ("manzana") al ordenador y a la empresa simplemente porque Jobs una vez había trabajado en una huerta.



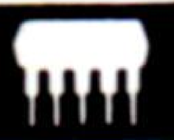
Durante 1975 y 1976 Wozniak se encerró en su garaje. Allí las ráfagas de inspiración de su genio lo impulsaban a trabajar de noche y de día, hasta que finalmente produjo el Apple II. Tenía 26 años. Aún hoy los expertos consideran la creación del Apple como una proeza sorprendente, por su diseño y sistema de circuitos tan brillantemente sencillos.

Afirman que Steve Wozniak lee el sistema de los diagramas de circuito y sincronización de un chip con la misma facilidad con que algunas personas leen el futuro en las hojas de té. Una de las innovaciones importantes del Apple II fue la simplificación de la unidad de disco. Antes de Wozniak ésta requería 30 chips, pero él la rediseñó para que sólo incluyera cinco en el caso del ordenador personal. No se trata tanto de que Wozniak hubiera creado algo completamente nuevo, sino que simplificó y comprimió tanto todos los componentes, que consiguió que pronto cualquiera pudiera usar un ordenador en su propia casa.

Wozniak nunca tuvo la intención de dedicarse a los negocios. Fue su socio Steve Jobs quien se encargó de vender los Apple y de crear la Apple Corporation. En la actualidad la empresa cuenta con 3 300 empleados en todo el mundo y se dice que gracias a ella al menos 50 personas se han hecho millonarias. Wozniak sólo posee el 4 % de las acciones de la Apple y jamás ha participado (ni le ha interesado hacerlo) en la gestión empresarial. Él sigue prefiriendo jugar con los ordenadores y soñar con ideas nuevas.

En el verano de 1983, después de un paréntesis de dos años durante los cuales se dedicó principalmente a organizar festivales de rock, Wozniak se reintegró en la Apple para trabajar en algunos proyectos nuevos. Nadie sabe con certeza en qué está trabajando. Al parecer está escribiendo algunos programas de aplicaciones. También está dirigiendo un proyecto del Apple II que aspira a crear un sistema personal de producción de video y edición, con gráficos de alta calidad, capaz de crear dibujos animados. Pero a Wozniak no sólo le interesa conseguir que las máquinas ya existentes sean mejores y más rápidas. También piensa que en el futuro se podrá diseñar un microordenador muy inteligente. Empleando solamente un sencillo programa este microordenador sería capaz de aprender cualquier materia.

El presidente
Steve Jobs, presidente de la Apple Computer Inc., era condiscípulo de Steve Wozniak y también su amigo y compañero de travesuras. Si Wozniak era el genio electrónico del equipo, Jobs era el niño precoz de los negocios. En 1975, cuando se diseñó el Apple I, Jobs tenía 20 años de edad. Él adivinó el potencial que ofrecía la venta de la máquina, entre las personas aficionadas a la informática, pero vio muy pronto que la podría introducir en un nuevo mercado, el de los ordenadores personales. Wozniak fue el creador del Apple, pero fue Jobs quien fabricó y vendió el aparato a nivel masivo. El Apple II se embaló en una caja, completo y listo para conectarlo en una casa particular, donde cualquiera pudiera usarlo. Mientras Wozniak y Jobs aún trabajaban en el taller del garaje, se invirtió en la empresa el primer capital: 1 350 dólares, provenientes de la venta de la furgoneta Volkswagen de Jobs y de la calculadora programable de Wozniak. Sin embargo, enseguida un joven millonario, Mike Markkula, prestó su apoyo financiero a lo que él consideraba un proyecto interesante. Cuando la existencia de la empresa se hizo pública, en 1980, fue tema de candente actualidad. De inmediato la compañía demostró su potencial al vender ordenadores por valor de 583 millones de dólares sólo en 1982.
El fabuloso crecimiento de la Apple Corporation ha sido ampliamente reconocido y recompensado en Estados Unidos. La revista *Time* le dedicó su portada a Steve Jobs, reconociéndolo como uno de los hombres de negocios más joven y de mayor éxito de Norteamérica



Varillas mágicas

Un lápiz óptico es un accesorio que sirve para dibujar o marcar en la pantalla. Vamos a explicar cómo funcionan estos extraordinarios dispositivos

La "aversión a los teclados" es una de las causas más comunes por las que la gente se muestra reacia a familiarizarse con los ordenadores, ya sea en su casa o en la oficina. Debido a que el teclado se asemeja al de una máquina de escribir y a que estas personas no saben mecanografía y, más aún, comprueban que el teclado contiene diversas teclas con signos desconocidos, temen hacer el ridículo. La solución a este problema es el lápiz óptico (junto a otros dispositivos tales como entrada de voz), que posee asimismo otras funciones.

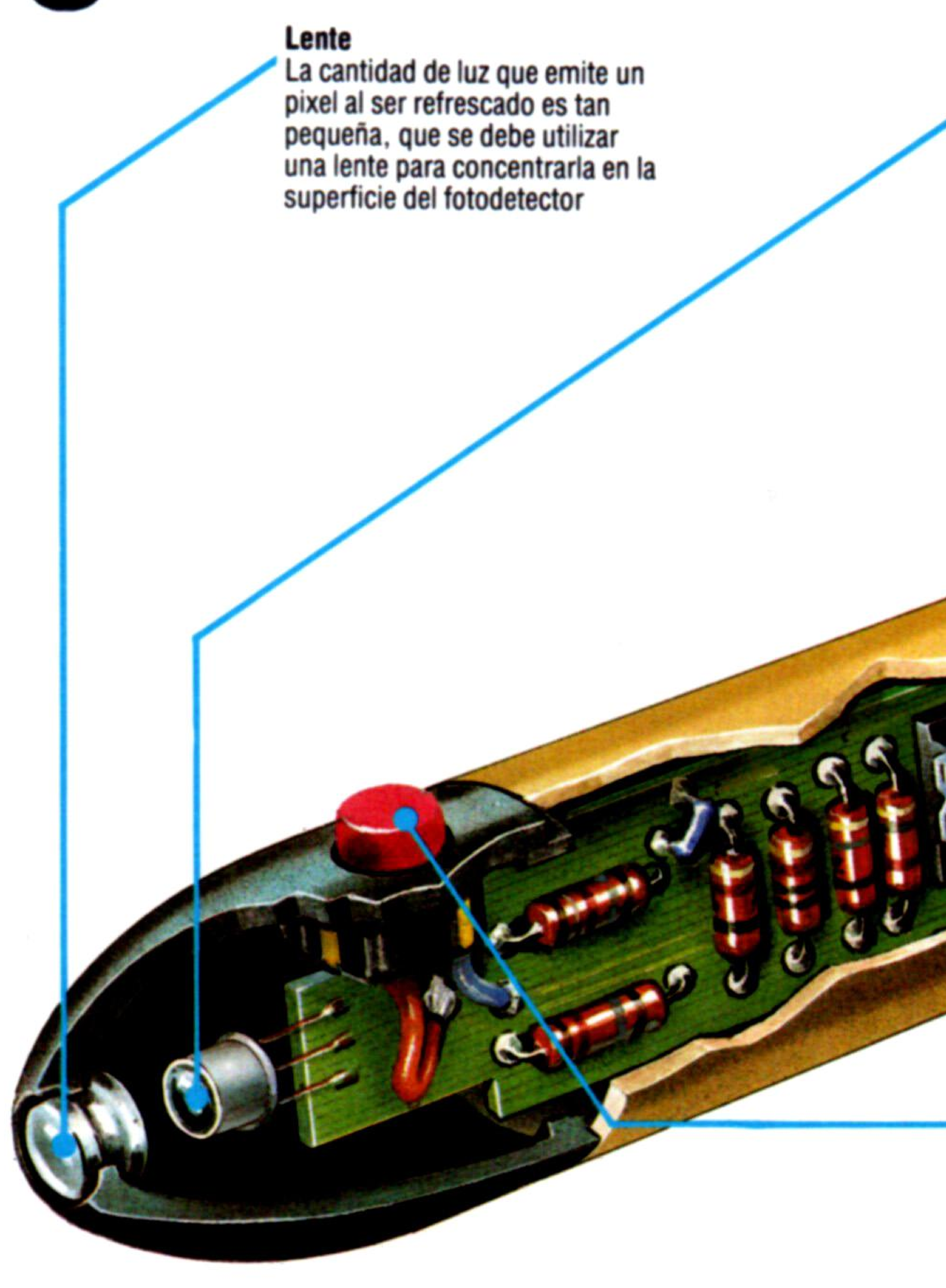
Un lápiz óptico es un dispositivo cilíndrico (su nombre deriva de su gran parecido con un lapicero corriente) de uno de cuyos extremos sale un cable en espiral similar al de los teléfonos. En el otro extremo de este cable hay un enchufe que se conecta en la parte posterior del ordenador. Cuando se apunta directamente a la pantalla con el lápiz óptico (en algunos ordenadores el lápiz ha de tocar ésta para activar un interruptor situado en su interior), el ordenador detecta la posición exacta que está señalando el lápiz.

Lo que sucede en realidad es que un fotodetector localizado en la punta del lápiz responde con un impulso eléctrico cuando pasa por el punto de luz que continuamente explora toda la pantalla para crear la imagen. El sistema de circuitos en el interior del chip controlador del video calcula dónde estaba el punto de exploración en el momento en que el lápiz óptico emitiera su señal.

El lápiz óptico, por lo tanto, se utiliza básicamente para seleccionar un ítem visualizado en pantalla. Sabiendo qué punto está señalando el lápiz, el ordenador deduce el carácter o el símbolo del que forma parte ese punto. Muchos programas de aplicaciones incorporan "menús" en su funcionamiento. Un menú es simplemente una lista de opciones que se visualiza en la pantalla, entre las cuales el usuario debe hacer su elección, como haría con el menú de un restaurante. En un programa de economía doméstica, el menú sería:

1) Efectuar un pago
2) Examinar el estado de cuentas bancario
3) Aceptar un recibo
   etcétera.

Normalmente el usuario debería indicar la acción que ha escogido pulsando una tecla (1, 2 o 3) o digitando una palabra de orden. Con un lápiz óptico, todo cuanto ha de hacer será señalar la opción requerida. El ordenador por lo general responde haciendo centellear esa opción determinada, para indicar que ha aceptado la entrada. Algunos programas más sofisticados funcionan casi completamente mediante este tipo de menús (se los suele describir como programas de opciones de trabajo), en cuyo caso el usuario sólo toca el teclado cuando se requiere una información verdadera, como por ejemplo el nombre y la dirección de alguien.



Rutinas especiales

La dificultad que suponen los programas de este tipo es que se deben escribir pensándolos especialmente para que trabajen con el lápiz óptico y no desde el teclado. En realidad la cuestión estriba en escribir una pequeña rutina que asuma las coordenadas que indican la posición normal del lápiz óptico desde el controlador de video y que determine cuál de las opciones se produce en la pantalla en esa posición. Lamentablemente, son pocos los fabricantes de software que proporcionan versiones de sus programas que puedan ejecutarse con lápiz óptico.

Sin embargo, además de utilizarse para seleccionar ítems, los lápices ópticos se pueden emplear para crear imágenes en pantalla. La mayoría de los ordenadores personales para los cuales existen lápices ópticos poseen un paquete de programas apto para tal fin. Al usuario se le presenta una pantalla en blanco (sobre la cual él puede dibujar un diagrama, un dibujo o un esbozo) y una sección separada (normalmente a lo largo de la parte inferior de la pantalla) que le proporciona una serie de funciones especiales para ayudarlo en el proceso de creación. Una de éstas puede ser una paleta de colores, que funciona igual que una paleta de óleos como la que emplean los pintores. El lápiz óptico se coloca sobre el próximo color requerido, y el lugar por donde después se pase el lápiz por la pantalla principal dejará una línea de ese color.

De la parte inferior de la pantalla, utilizada como paleta, el usuario también puede seleccionar distintas

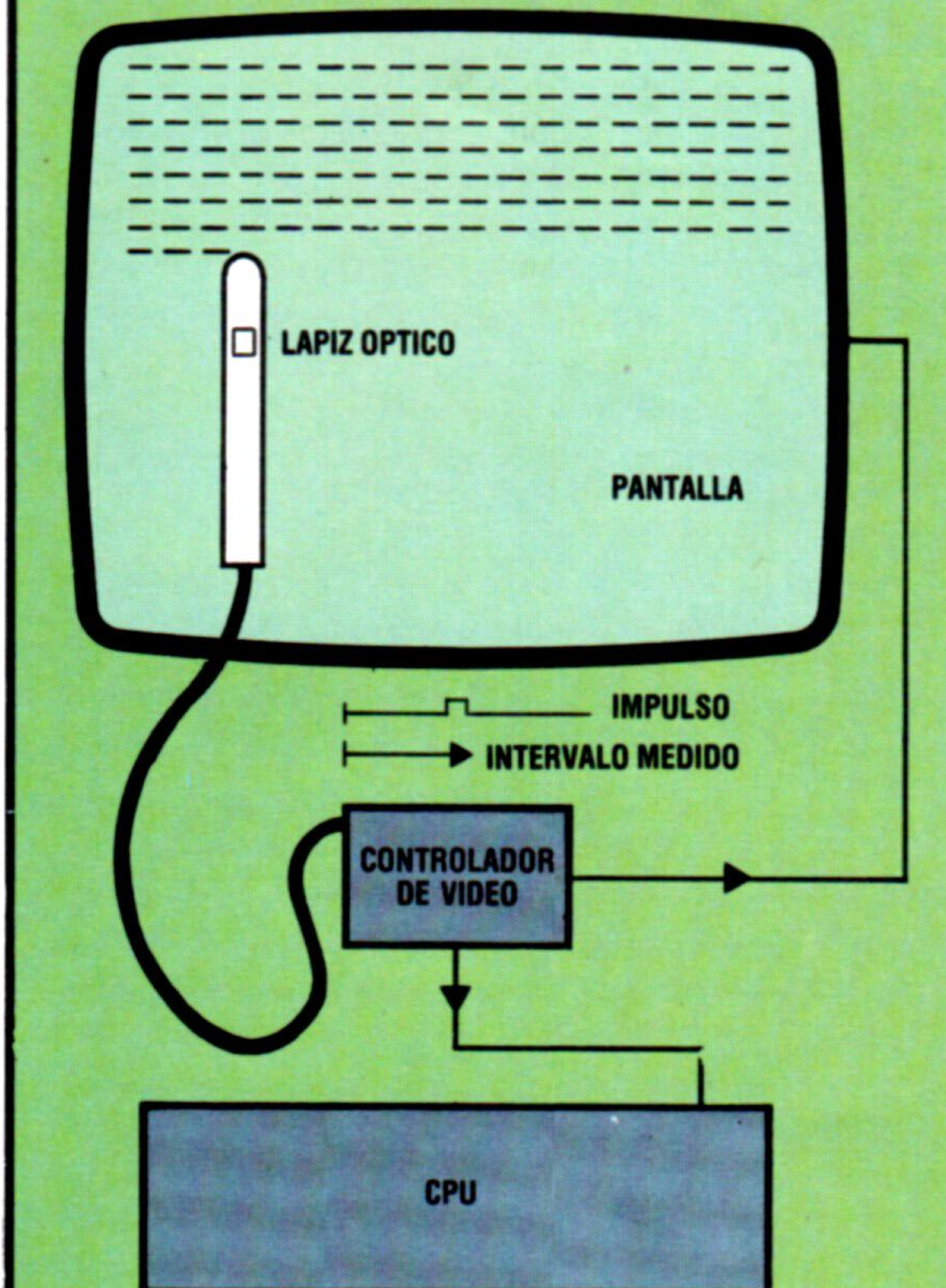
Fotodetector
Se trata de un dispositivo semiconductor que, por decirlo de alguna manera, es como un transistor o un diodo al que se le hubiera aserrado la punta. La luz que recibe este dispositivo controla la corriente eléctrica que fluye a través del mismo

Kai Choi

Circuitos de amplificación
Éstos sirven para detectar y amplificar la corriente que pasa a través del detector y para enviarle una señal de retorno adecuada al chip controlador de video del ordenador. Algunas veces estos circuitos están alojados fuera del lápiz propiamente dicho

Interruptor
La mayoría de los lápices ópticos incorporan alguna clase de interruptor, a veces para operar manualmente, pulsándolo con un dedo, y otras veces para activar presionando el lápiz óptico contra la pantalla. El interruptor es necesario para que el lápiz no reaccione a la luz (por ejemplo, a la luz de la habitación) cuando no se lo está utilizando para seleccionar un ítem en la pantalla

Cable
Este cable en espiral al estilo de los del teléfono se enchufa directamente en la parte posterior del ordenador y desde allí conecta con el chip controlador de video



Cómo funciona
El funcionamiento del lápiz óptico suele depender del chip controlador de video situado en el interior del ordenador. Este chip, explorando la pantalla muchas veces por segundo con líneas horizontales, produce imágenes de video en el televisor o en el monitor (esta acción se denomina *raster scan*). Los televisores españoles tienen 625 líneas en la pantalla y la frecuencia de exploración es de 50 por segundo. La salida del lápiz óptico también se deriva directamente a este chip. Cuando se señala un ítem de la pantalla con el lápiz óptico y se activa el interruptor, el controlador de video se limita a medir el tiempo que media desde la última vez que comenzó a trazar las líneas en la parte superior de la pantalla y el instante en que recibió el impulso proveniente del lápiz óptico. A partir de esta información calcula sobre qué línea está situado el lápiz y a qué distancia se halla esta línea. Esta información se convierte luego en las coordenadas X e Y que necesita el programa

Liz Dixon

anchuras de trazo y texturas y, asimismo, puede dibujar círculos y cuadrados. En resumen, todo cuanto exponíamos en "El artista electrónico" (véase p. 26) se puede realizar con un lápiz óptico y, probablemente, mucho más rápido que mediante el teclado.

Asimismo, están comenzando a aparecer en el mercado juegos que emplean lápices ópticos. Capturar monstruos extraterrestres es mucho más fácil con un lápiz óptico que con el teclado, de manera que los juegos han de incorporar otros grados de dificultad. Otros juegos más sosegados, como el ajedrez informatizado, también se pueden beneficiar de la utilización del lápiz óptico: uno señala adónde quiere desplazar el caballo y el ordenador se ocupa de todo lo demás.

Quizá el grupo más numeroso de usuarios de lápices ópticos sea, no obstante, el de ingenieros y diseñadores. Los sistemas de CAD (Computer Aided Design: diseño auxiliado por ordenador) se emplean bastante en los anuncios publicitarios de nuevos modelos de coches; en realidad, son sistemas informatizados como tantos otros, pero con un software especializado y gráficos de gran calidad. Si se utiliza un sistema CAD para diseñar un nuevo dispositivo electrónico, la pantalla incorporará la representación de todos los componentes a los que el diseñador pueda desear acceder; éste puede "escoger" cualquiera de ellos y colocarlo en el lugar de la pantalla que crea conveniente.

El lápiz óptico constituye uno de los mejores ejemplos de accesorio para ordenador que, además de poseer gran valor práctico, resulta fácil y divertido de emplear.

Planificando el futuro

	A	B		N
1	ENERO	FEBRERO	1	TOTALES
2	42,41	18,75	2	388,4
3	160,35	149,89	3	1732,7
4			4	

Campos de juego
Una hoja electrónica está dividida en líneas y columnas; la intersección de una línea con una columna se denomina *campo* o *celda* y se puede direccionar a partir de sus coordenadas (A1, B3, etc.). Cada campo puede contener un título (p. ej., ENERO), un número (p. ej., 149,89) o una fórmula. El campo N2 contiene la fórmula 'SUM(A2: M2)', que es la suma de las cifras de la línea superior, de enero a diciembre. El resultado de este cálculo está visualizado: 388,4. Observe que se ha dividido la hoja electrónica en dos ventanas, de manera que las cifras de marzo-diciembre no aparecen visibles

Un programa de "hoja electrónica" puede ayudar a darle a su ordenador una utilidad lucrativa. Sus principales aplicaciones son planificación, presupuesto y previsión financiera

Se ha calculado que los gerentes dedican aproximadamente el 30 % de su tiempo a la elaboración de presupuestos, una actividad que siempre implica considerar las respuestas a muchas preguntas del tipo "¿qué ocurriría si...?" Tradicionalmente se utilizaba para ello una hoja de papel, por lo general equivalente al tamaño de una doble página de *Mi Computer*. En ella se trazaban verticalmente una docena o más de columnas, cada una de las cuales se encabezaba con el número de un mes, y todos los tipos de gastos uno debajo del otro sobre un costado. En cada columna se incluían los gastos mensuales en las diversas categorías. Siguiendo este método, uno podía sumar cada columna para obtener los gastos totales mes por mes, y sumar individualmente todas las líneas para hallar el total gastado durante un año entero en cada rubro.

El problema se planteaba cuando uno había planificado gastar demasiado (o, peor aún, muy poco) y debía volver atrás y alterar una gran cantidad de cifras, y volver a calcular en función de ellas los totales de las líneas y las columnas.

Utilizando un programa de hoja electrónica, uno puede volver a calcular la página entera de cifras cada vez que se altera un único elemento básico. Si, por ejemplo, ha variado el costo del transporte del mes de enero, esta variación alterará los gastos totales de ese mes; así como el total de los gastos correspondientes a ese rubro, con sólo tocar una tecla. ¡No hay que sorprenderse de que los paquetes de hojas electrónicas sean el tipo de programas que más se venden en todo el mundo!

Al igual que la mayoría de los programas comerciales, las hojas electrónicas por lo general se escriben con *overlays*, es decir, que en realidad no todo el programa reside realmente en la máquina todo el tiempo. Si se considera que el programa está dividido en subrutinas (véase p. 77), la subrutina que no se requiera en la operación normal no será recuperada del almacenamiento de apoyo (disco o cinta) hasta que se la "llame". El sistema operativo *colocará* entonces esa subrutina *sobre* sobre la que se ha vuelto redundante (de ahí el nombre *overlay*: sobreponer). Como se puede imaginar, este método para ampliar la memoria disponible es verdaderamente muy útil, pero implica que a menudo uno debe esperar mientras la información se transfiere desde el medio de almacenamiento de apoyo a la memoria principal.

Los paquetes de hojas electrónicas (por lo general se los puede reconocer por su nombre, ya que la mayoría de las veces terminan con la partícula *calc*) existen para una gran variedad de ordenadores personales y de gestión empresarial. El más popular en Gran Bretaña, en función de su volumen de ventas, es el Visicalc, escrito originalmente para el Apple II y que salió a la venta a mediados de 1979. El mercado de software para microordenadores reacciona con mucha rapidez ante el éxito obtenido por uno de sus productos, y este caso no fue una excepción. Antes de que uno pronunciara las palabras "hoja electrónica", ya se hallaban a la venta, a lo largo de todo el país, programas para cualquier clase de máquina, apropiados o no.

Para que una hoja electrónica sea realmente útil ha de satisfacer dos requisitos: dimensiones (no necesariamente las que se ven en la pantalla, porque, como veremos más adelante, lo que se contempla es sólo una "ventana" de la totalidad) y una buena gama de órdenes para control y formato.

Esto significa que existen severas limitaciones en cuanto a las clases de máquinas que tienen posibilidades de ejecutar un programa de hoja electrónica aprovechándolo al máximo. Por regla general, las exigencias mínimas son 32 Kbytes de RAM y una pantalla de 80 caracteres en el caso de una aplicación de gestión, si bien una pantalla de 40 caracteres probablemente será suficiente si se trata de fines domésticos.

Los usuarios de ordenadores personales comprobarán que muchos de los paquetes disponibles para máquinas basadas en cassette, como el Spectrum y el Vic-20, si bien por su propia naturaleza son de dimensiones y potencial limitados, también resultarán muy útiles.

Dado que las hojas electrónicas poseen la capacidad de responder a preguntas del tipo "¿qué ocurriría si...?", es obvio que se pueden emplear para elaborar modelos simples informatizados (véase p. 101). Allí dábamos un ejemplo del trabajo del analista de sistemas; si se diera el caso de que usted utilizara un paquete de hoja electrónica, comprendería enseguida la evidente necesidad de efectuar una planificación similar e igualmente cuidadosa. Hemos comentado que las bases de datos consisten en una cantidad de información no clasificada que se ordena de acuerdo a los requerimientos del usuario cuando se recupera la información. Existen procesadores de textos (otro software de gran demanda) que le permiten al usuario desplazar palabras o bloques enteros de texto en el momento de ejecución. Pero las hojas electrónicas son un poco diferentes, en el sentido de que en realidad exigen por parte del usuario un proceso de planificación.

La línea inferior
El cursor es el bloque rectangular que normalmente ocupa el campo N2. Si se digita algo, lo digitado aparecerá en el campo donde estaba situado el cursor. El contenido completo de ese campo también se visualizará en la línea de órdenes de la hoja electrónica, que en este caso está en la parte inferior de la página

FORMULA: SUM (A2:M2)

Por ejemplo, si está utilizando una hoja electrónica para analizar sus gastos, quizá le interese agrupar todos los gastos relacionados con la vivienda (pagos de alquiler o de hipoteca, impuestos, seguro, etc.) y luego emplear el resultado de ese cálculo en una tabla mayor dentro de la misma hoja. Habrá de ser muy cuidadoso al sumar todos los gastos relacionados con la casa antes de trasladar ninguna cifra a la tabla mayor.

Cada dirección individual de la hoja electrónica, denominada celda, se direcciona y localiza en función de sus coordenadas X e Y. Horizontalmente se utilizan las letras A - Z, AA - AZ, e incluso BA - BM, para cubrir la anchura total posible de la hoja, que en las versiones más corrientes es de 65 celdas. Verticalmente se pueden emplear los números del 0 al 255. Cada celda puede contener una etiqueta (por ejemplo "ventas" o "ganancia"), un valor al que se le da entrada como resultado de un cálculo, o bien derivado del mismo (como 1 000), o la fórmula para efectuar ese cálculo, como B4 + B6 * B5. Las fórmulas, debido a que suelen exceder las dimensiones de visualización del recuadro, normalmente se visualizan en una línea separada en la parte superior de la pantalla. Al comenzar una hoja electrónica nueva las dimensiones de la celda estarán preestablecidas, quizás en ocho o nueve dígitos o caracteres. Esto se conoce como dimensiones por omisión. Por lo general, es posible acortar o alargar las celdas para adaptarlas al tipo de cálculo que se esté realizando. Algunos paquetes permiten que la columna situada más hacia la izquierda (normalmente sus títulos o descripciones) sea más ancha que las demás. Y no es necesario decidir de inmediato el tamaño que se desea darles a las celdas. La mayoría de las versiones son susceptibles de ampliar o reducir, incluso después de haber colocado información en ellas. En caso de que se redujeran las dimensiones por debajo de la longitud del contenido, la parte no visualizada no se borra sino que solamente queda oculta a la vista.

El último de los componentes principales de la hoja es la línea de orden, que aparece en la parte superior o inferior de la pantalla en respuesta a la tecla de "orden": /, por ejemplo. Estas órdenes se emplean para dar su formato y manipular el trazado de la hoja en sí misma, no de la información, si bien puede llegar a incidir en la forma en que se presenta ésta. La mayor parte de las versiones más populares de programas para hojas electrónicas permiten efectuar una gran variedad de operaciones con la base de datos. Se puede, por ejemplo, borrar, trasladar o copiar líneas o columnas enteras, y además dividir la ventana para que se visualicen juntas partes de la hoja que normalmente están muy alejadas entre sí como para aparecer juntas en la ventana; después es posible también hacer mover en espiral (mover a través) estas ventanas por separado.

El movimiento entre las celdas se suele realizar mediante la tecla de control del cursor; no obstante, otra tecla de orden permite saltar hasta otra celda determinada. Cargar y guardar, borrar y proteger, todo ello se efectúa empleando las teclas de órdenes, y en este punto vale la pena volver a recalcar la importancia de guardar regularmente el trabajo que uno haya realizado. Componer una hoja electrónica es una tarea que toma mucho tiempo, por lo regular mucho más que el que se necesita para dar entrada a los datos. Por regla general, *siempre* se ha de guardar una hoja antes de empezar a dar entrada a los datos. Luego, en el caso de que se incurriera en un error garrafal, el haberla conservado le evitará una pérdida irreparable.

Los resultados se transmiten a la impresora empleando una tecla de orden, pero el usuario ha de asegurarse de definir la parte de la hoja que desea imprimir mediante la utilización de los parámetros. Así como la pantalla es una ventana, o un fragmento de la totalidad de la hoja, la misma función desempeñará la hoja impresa a través de una impresora de salida. Si desea imprimir una hoja más ancha que su impresora, lo aconsejable es hacer dos impresiones y luego unir las dos páginas.

Como ya hemos apuntado, el empleo de ventanas permite que se visualicen simultáneamente dos partes diferentes de la hoja. También se pueden imprimir las hojas con ventana dividida. Esto es especialmente útil al dar entrada a la información, puesto que uno puede hacer una referencia a una entrada anterior. La mayoría de los paquetes le permiten al usuario "retener" la línea superior o la primera línea, o ambas a la vez, que él crea por algún motivo, tomando en cuenta que por lo general éstas contienen los títulos o etiquetas.

Hasta este momento hemos venido considerando la hoja como una tabla a la cual uno sólo puede acceder en serie (un ítem después del otro, a lo largo de una línea o de una columna); pero si uno no desea seguir el trazado normal, sumando líneas o columnas, no existe ningún impedimento para trazar la hoja de cualquier otra forma si procediendo de esa manera se facilita la

¿Qué ocurriría si...?
Si se modifica el contenido de cualquier campo, la hoja electrónica volverá a calcular automáticamente todos los campos que dependan de esa cifra. La velocidad y sencillez de este proceso estimula al usuario a verificar la validez de sus proyectos, para comprobar lo que sucedería con los beneficios globales si variaran determinadas condiciones. Tomemos el ejemplo de un vendedor de frutas...



Si el precio de la gasolina aumentara en un X %... Luego los costes mensuales de transporte se incrementarían en un Y %... Esto, a su vez, haría subir el precio al por mayor de las manzanas... Y ello incidiría directamente en el consumidor a través del aumento de los precios...

Kevin Jones

Mark Watkinson

fluidez de pensamiento del usuario. No obstante, conviene recordar que este tipo de sofisticación dentro de la base de datos requerirá un análisis de los sistemas mucho más exhaustivo que el que normalmente se lleva a cabo.

Las hojas electrónicas para gestión empresarial, como Visicalc, Supercalc y Masterplan, le ofrecen al usuario la posibilidad de pasar información desde la hoja electrónica a paquetes de gestión para bases de datos o tratamiento de textos, y existen muchos programas complementarios para que las salidas se realicen en una variedad de formas gráficas: circulares, por ejemplo, o de barras.

Ya hemos mencionado una de las posibles aplicaciones de las hojas electrónicas para el usuario de ordenadores personales: el análisis de los gastos y el presupuesto del hogar. Otra aplicación muy adecuada para el hogar sería en la instalación de la calefacción central, para la cual se han de considerar gran cantidad de variables: el tipo de combustible que se empleará, la cantidad y la clase de radiadores, la salida de la caldera, etc. De hecho, utilizar una hoja electrónica es de gran ayuda para cualquier proceso que implique tomar una serie de decisiones, en especial porque el usuario está casi obligado a contemplar todas y cada una de las posibilidades.

Quizá la característica más notable de una hoja electrónica para gestión empresarial ejecutada en una máquina adecuada, sea su asombrosa velocidad de funcionamiento. Ésta es una función de la programación en código de lenguaje máquina y no es nada sorprendente que la velocidad de un paquete escrito en BASIC para un pequeño ordenador personal pueda ser, en comparación, motivo de alarma.

Tal vez resulte interesante considerar algunos de los problemas con los que uno se podría enfrentar si intentara escribir un programa de este tipo en BASIC, aunque sólo fuera como un medio para llegar a comprender la complejidad que entrañaría dicha tarea.

Para empezar, cada celda se habría de definir de tres maneras. La celda debería ser capaz de retener datos de "variables alfanuméricas", como "enero" o "tarifas"; debería ser capaz de retener datos numéricos para su utilización en operaciones aritméticas, como por ejemplo, el total de las cuotas pagadas en enero; y también podría ser necesario que retuviera una fórmula que, en esencia, fuera una línea del código de programación, como "cuotas anuales/12" para obtener la cuota mensual. Luego las dimensiones de cada celda deberían ser susceptibles de ampliarse y reducirse, pero sin perder su parte menos significativa, para lo cual todas ellas deberían duplicarse: una aparecería en la pantalla y la otra, que siempre retendría toda la información, quedaría oculta dentro del programa.

Como se puede apreciar, sólo la manipulación de la información es en sí misma una tarea complicada; y recuerde que los paquetes más sofisticados ¡pueden constar de hasta 16 000 celdas individuales! Las técnicas que se utilizan para escribir esta clase de programas se parecen mucho a las que se emplean para escribir intérpretes para lenguajes como el BASIC o el FORTH, y también se hace uso de técnicas similares para el software de gestión por base de datos.

Todo esto viene al caso para explicar el porqué del elevado costo de estos programas empresariales escritos con una finalidad concreta. Un paquete como el Visicalc o el Supercalc puede ser muy caro, pero a la hora de contemplar la posibilidad de comprarlo, quizá el punto a considerar sea el ahorro que supone su utilización. La relativamente sencilla aplicación que realizamos anteriormente, la del director que recopilaba presupuestos, por ejemplo, puede representar un ahorro de quizá entre un 15 y un 20 % en un solo año. El costo del software constituye una parte insignificante en relación al monto total de inversión sin contar el ahorro de tiempo y energía que trae aparejado su aplicación. Por cierto, el paquete Visicalc probablemente haya sido el mejor amigo de los agentes de ventas del Apple.

Ventanas al mundo
Así como podemos desplazar el cursor por la pantalla hacia la derecha o hacia la izquierda, hacia arriba o hacia abajo, mediante el control del teclado, en una hoja electrónica podemos mover la pantalla a través de la hoja. Esta sofisticación permite visualizar una superficie mucho mayor que la de la pantalla.
Algunos ordenadores portátiles con pantalla de dimensiones limitadas, como el Epson HX-20 y el Osborne-1, utilizan el mismo sistema para todas sus operaciones

	A	B	C	D	E	F	G	H	I
1	TITULOS	ENERO	FEBRERO	MARZO	ABRIL	MAYO	JUNIO	JULIO	AGOSTO
2	ALQUILER	180,00	180,00	180,00	192,00	192,00	192,00	192,00	192,00
3	TASAS								51,90
4	SEGURO								
5	TOTAL								--------
6	FAMILIA								243,90
7	TRANSPORTE								,00
8	COMIDA								,00
9	ROPA								,00
10	LIBROS								,00
11	TOTAL								--------
12	ESCUELA								,00
13	GASOLINA								92,00
14	REVISION								,00
15	OBRAS								18,00
16	IMPUESTOS								,00
17	SEGURO								,00
18	KILOMETRAJE								900,00
19	TOTAL								--------
20	COCHE								110,00
21	VARIOS								,06
22	VIAJES								74,00
23	ALOJAMIENTO								340,00
24	COMIDA								190,00
25	DIVERSIONES								38,00
26	TOTAL								--------
27	OCIO								642,06

	D	E	F	G	H
8	8.00	5.20	8.00	8.00	4.00
9	6.50	2.25	.00	18.00	3.40
10	.00	.00	14.50	.00	6.95
11	--------	--------	--------	--------	--------
12	16.80	8.95	25.70	29.20	15.95
13	34.00	41.00	48.00	35.00	39.00
14	.00	112.00	.00	.00	44.00
15	.00	.00	.00	.00	.00
16	.00	.00	.00	.00	.00
17	.00	.00	.00	.00	182.00
18	600.00	950.00	1200.00	620.00	820.00
19	--------	--------	--------	--------	--------
20	34.00	153.00	48.00	35.00	265.00
21	.06	.16	.04	.06	.32
22	.00	.00	.00	.00	.00
23	.00	.30	.03	.00	.00
24	72.00	36.00	41.00	18.00	46.00
25	13.00	22.00	14.00	13.00	22.00
26	--------	--------	--------	--------	--------
27	85.06	58.16	55.04	31.06	68.32

```
H12           Form=H7+H8+H9+H10
Width:  9  Memory: 20 Last Col/Row:027     ? for HEL
1>_
```

DONDE CONSEGUIR TU sinclair

ALAVA
COMPONENTES ELECTRONICOS GAZTEIZ
Domingo Beltran, 58 (Vitoria)
DEL CAZ
Avda. Gazteiz, 58 (Vitoria)
VALBUENA
Virgen Blanca, 1 (Vitoria)

ALBACETE
ELECTRO MIGUEL
Tesifonte Gallego, 27
TECON
Maria Marin, 13

ALICANTE
ASEMCA (Villena)
Avda. de la Constitucion, 54 (Villena)
CONSULTING DESARROLLO INFORMATICO
Pais Valencia, 54 (Alcoy)
COMPONENTES ELECTRONICOS LASER
Jaime M.ª Buch, 7
ELECTRODATA LEVANTE
San Vicente, 28
ELECTRONICA AITANA
Limones, s/n. Edificio Urgull (Benidorm)
ELECTRONICA OHMIO
Avda. El Hamed, 1
LIBRERIA LLORENS
Alameda, 50 (Alcoy)

AVILA
FELIX ALONSO
San Segundo, 15

BADAJOZ
MECANIZACION EXTREMEÑA
Vicente Barantes, 18
SONYTEL
Villanueva, 16

BARCELONA
ARTO
C/ Angli, 43
BERENGUERAS
C/ Diputacion, 219
CATALANA D'ORDINADORS
C/ Trafalgar, 70
CECSA
C/ Mallorca, 367
COMPUTERLAND
C/ Infanta Carlota, 89
COMPUTERLAND
Trav. de Dalt, 4
COPIADUX
C/ Dos de Mayo, 234
D. P. 2000
C/ Sabino de Arana, 22-24
DIOTRONIC
C/ Conde Borrell, 108
EL CORTE INGLES
Avda. Diagonal, 617-619
EL CORTE INGLES
Pza. Cataluña, 14
ELECTRONICA H. S.
C/ S. Jose Oriol, 9
ELECTRONICA SAUQUET
C/ Guilleries, 10
ELEKTROCOMPUTER
Via Augusta, 120
EXPOCOM
C/ Villarroel, 68
GUIBERNAU
C/ Sepulveda, 104
INSTA-DATA
P.º S. Juan, 115
MAGIAL
C/ Sicilia, 253
MANUEL SANCHEZ
Pza. Major, 40 (Vic)
MILLIWATTS
C/ Melendez, 55 (Mataró)
ONDA RADIO
Gran Via, 581
RADIO ARGANY
C/ Borrell, 45
RADIO SONDA
Avda. Abad Margat, 77 (Tarrasa)
RAMEL ELECTRONICA
Cr. de Vic, 3 (Manresa)
REDISA GESTION
Avda. Sarria, 52-54
RIFE ELECTRONICA
C/ Aribau, 80, 5.º, 1.ª
SERVICIOS ELECTRONICOS VALLES
Pza. del Gas, 7 (Sabadell)
SISTEMA
C/ Balmes, 434
S. E. SOLE
C/ Muntaner, 10
SUMINISTROS VALLPARADIS
C/ Dr. Ferren, 172 (Tarrasa)
TECNOHIFI, S. A.
C/ La Rambleta, 19
VIDEOCOMPUT
P.º Pep Ventura, 9, Bl. C, Bjos. Bis (Vic)

BURGOS
COMELECTRIC
Calzada, 7
ELECTROSON
Conde don Sancho, 6

CACERES
ECO CACERES
Diego Maria Crehuet, 10-12

CADIZ
ALMACENES MARISOL
Camoens, 11 (Ceuta)
INFORSA
Avda. Fuerzas Armadas, 1 (Algeciras)
ELECTRONICA VALMAR
Ciudad de Santander, 8
M. R. CONSULTORES
Multi Centro Merca 80 (Jerez de la Frontera)
PEDRO VAREA
Porvera, 36 (Jerez de la Frontera)
LEO COMPUTER
Garcia Escamez, 3
SONYTEL
Queipo de LLano, 17
SONYTEL
Jose Luis Diez, 7
T. L. C. Y AUTOMATICA
Dr. Herrera Quevedo, 2

CASTELLON
NOÙ DESPACH'S
Rey D. Jaime, 74

CIUDAD REAL
COMERCIAL R. P.
Travesera de Coso, 2 (Valdepeñas)
ECO CIUDAD REAL
Calatrava, 8

CORDOBA
ANDALUZA DE ELECTRONICA
Felipe II, 15
CONTROL
Conde de Torres Cabrera, 9
ELECTRONICA PADILLA
Sevilla, 9
MORM
Plaza Colón, 13
SONYTEL
Arte, 3
Avda. de los Mozárabes, 7

CUENCA
SONYTEL
Dalmacio Garcia Izcara, 4

GERONA
AUDIFILM
C/ Albareda, 15
CENTRE DE CALCUL DE CATALUNYA
C/ Barcelona, 35
S. E. SOLE
C/ Sta. Eugenia, 59

GRANADA
INFORMATICA Y ELECTRONICA
Melchor Almagro, 8
SONYTEL
Manuel de Falla, 3
TECNIGAR
Ancha de Gracia, 11

GRANOLLERS
COMERCIAL CLAPERA
C/ Maria Maspons, 4

GUIPUZCOA
ANGEL IGLESIAS
Sancho el Sabio, 7-9
BHP NORTE
Ramon M.ª Lili, 9
ELECTROBON
Reina Regente, 4

HUELVA
SONYTEL
Ruiz de Alda, 3

HUESCA
ELECTRONICA BARREU
M.ª Auxiliadora, 1

IBIZA
IBITEC
C/ Aragon, 76

JAEN
CARMELO MILLA
Coca de la Piñera, 3
MARA ILUMINACION
Avda. Linares, 13 (Ubeda)
MICROJISA
Garcia Rebull, 8
SONYTEL
José Luis Diez, 7
SONYTEL
Pasaje del Generalisimo, 3 (Linares)

LA CORUÑA
DAVIÑA
Republica de El Salvador, 29 (Santiago)
PHOTOCOPY
Teresa Herrera, 9
SONYTEL
Avda. de Arteijo, 4
SONYTEL
Tierra, 37

LAS PALMAS DE GRAN CANARIA
COMPUTERLAND
Carvajal, 4
CHANRAI
Triana, 3
EL CORTE INGLES
Jose Mesa y Lopez, 18

LEON
ELECTROSON
Avda. de la Facultad, 15
MICRO BIERZO
Carlos I, 2 (Ponferrada)
RADIO RACE
Modesto Lafuente, 3

LERIDA
SELEC
C/ Ferrer y Busquet, 14 (Mollerusa)
SEMIC
C/ Pi y Margall, 47

LUGO
ELECTROSON
Concepcion Arenal, 38
SONYTEL
Primo de Rivera, 30

MADRID
ALFAMICRO
Augusto Figueroa, 16
BELLTON'S
Torpedero Tucuman, 8
CHIPS-TIPS
Pto. Rico, 21
CMP
Pto. Santa Maria, 128
COMPUTERLAND
Castello, 89
COSESA
Barquillo, 25
DINSA
Gaztambide, 4
DISTRIBUIDORA MADRILEÑA
Todos sus centros
ELECTROSON
Duque de Sexto, 15 (y otros centros)
INVESMICROSTORE
Genova, 7
J.P. MICROCOMPUT
Montesa, 44
EL CORTE INGLES
Todos sus centros
ELECTRONICA SANDOVAL
Sandoval, 4
PENTA
Dr. Cortezo, 12
RADIO CINEMA
Antonio Acuña, 3
RADIO QUER
Todos sus centros
SONYTEL
Clara del Rey, 24 (y todos sus centros)
SONICAR
Vallehermoso, 19
VIDEOMUSICA
Orense, 28

MALAGA
EL CORTE INGLES
Prolongación Alameda, s/n.
INGESCON
Edificio Galaxia
SONYTEL
Salitre, 13

MELILLA
OFI-TRONIC
Hermanos Cayuela, 11

MENORCA
ELECTRONICA MENORCA
C/ Miguel de Veri, 50 (Mahon)

MURCIA
COMPUTER LIFE
Alameda San Anton, 2 (Cartagena)
EL CORTE INGLES
Libertad, 1
ELECTRONICA COMERCIAL CRUZ
Rio Segura, 2
MICROIN
Gran Via, 8

NAVARRA
ENER
Paulino Caballero, 39
GABINETE TECNICO EMPRESARIAL
Juan de Labrit, 3
JOSE LUIS DE MIGUEL
Arrieta, 11 bis

OVIEDO
AUTECA
Valentin Masip, 25
EDIMAR
Cangas de Onis, 4-6 (Gijon)
ELECTRONICA RATO
Versalles, 45 (Aviles)
RADIO NORTE
Uria, 20
RESAM ELECTRONICA
San Agustin, 12 (Gijon)
RETELCO
Cabrales, 31 (Gijon)
SELECTRONIC
Fermin Canellas, 3

ORENSE
SONYTEL
Concejo, 11

PONTEVEDRA
EL CORTE INGLES
Gran Via, 25 (Vigo)
ELECTROSON
Santa Clara, 32
ELECTROSON
Venezuela, 32 (Vigo)
SONYTEL
Salvador Moreno, 27
SONYTEL
Gran Via, 52 (Vigo)
TEFASA COMERCIAL
San Salvador, 4 (Vigo)

PALMA DE MALLORCA
GILFT
Via Alemania, s/n
IAM
C/ Cecilio Metlo, 5
TRON INFORMATICA
C/ Juan Alcover, 54, 6.º C

LA RIOJA
YUS COMESSA
Cigueña, 15

SALAMANCA
DEL AMO
Arco, 5
PRODISTELE
España, 65

SANTANDER
LAINZ S. A.
Reina Victoria, 127
RADIO MARTINEZ
Dr. Jimenez Diaz, 13

SEGOVIA
ELECTRONICA TORIBIO
Obispo Quesada, 8

SEVILLA
A.D.P.
San Vicente, 3
EL CORTE INGLES
Duque de la Victoria, 10
SCI
Aceituno, 8
SONYTEL
Pages del Corro, 173
Adriano, 32

TARRAGONA
AIA
Rambla Nova, 45, 1.º
CIAL. INFORMATICA TARRAGONA
C/ Gasometro, 20
ELECTRONICA REUS
Avda. Prat de la Riba, 5 (Reus)
SEIA
Rambla Vella, 7 B
S.E. SOLE
C/ Cronista Sese, 3
T. V. HUGUET
Pza. Major, 14 (Montblac)
VIRGILI
C/ Dr. Gimbernat, 19 (Reus)

STA. CRUZ DE TENERIFE
COMPUTERLAND
Mendez Nuñez, 104 B
TRENT CANARIAS
Serrano, 41

VALENCIA
ADISA
San Vicente, 33 (Gandia)
CESPEDES
San Jacinto, 6
COMPUTERLAND
Marques del Turia, 53
DIRAC
Blasco Ibañez, 116
EL CORTE INGLES
Pintor Sorolla, 26
Melendez Pidal, 15
PROMOCION INFORMATICA
Pintor Zariñena, 12

VALLADOLID
SONYTEL
Leon, 4

VIZCAYA
BILBOMICRO
Aureliano del Valle, 7
DATA SISTEMAS
Henao, 58
DISTRIBUIDORA COM
Gran Via, 19-21 y todos sus centros
EL CORTE INGLES
Gran Via, 9
ELECTROSON
Alameda de Urquijo, 71
San Vicente, 18 (Baracaldo)
GESCO INFORMATICA
Alameda de Recalde, 76
KEYTRON
Hurtado de Amezaga, 20

ZAMORA
MEZZASA
Victor Gallego, 17

ZARAGOZA
EL CORTE INGLES
Sagasta, 3
SONYTEL
Via Pignatelli, 29-31



DISTRIBUIDOR EXCLUSIVO:
INVESTRONICA

Central Comercial
TOMAS BRETON, 60
TELF. 468 03 00
TELEX 23399 IYCO E
MADRID

Delegación Cataluña
MUNTANER, 565
TELF. 212 68 00
BARCELONA




10009
9 788485 822836